The Dissertation Committee for Mehmet Ferhat Candas
certifies that this is the approved version of the following dissertation:

Considering inventory in service parts logistics network design problems with time-based service constraints

Committee:

Erhan Kutanoglu, Supervisor

Anant Balakrishnan

John Hasenbein

David Morton

Wilbert Wilhelm

Considering inventory in service parts logistics network design problems with time-based service constraints

by

Mehmet Ferhat Candas, B.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2007

Dedicated to *LOVE*.

Acknowledgments

This dissertation could not have been written without the invaluable help of my advisor, Dr. Erhan Kutanoglu. I am forever indebted to Dr. Kutanoglu for his expertise, patience and his seemingly endless commitment to me and to my research. I am especially grateful to the members of my dissertation committee - Dr. Anant Balakrishnan, Dr. John Hasenbein, Dr. David Morton and Dr. Wilbert Wilhelm - for their contributions to my professional and academic accomplishments. I must thank my family and friends who have patiently endured and supported me through every step of this journey. Finally, I would like to acknowledge the National Science Foundation for funding our research, IBM for providing data and insights, and also Dash Optimization for allowing us to use Xpress-MP solver for this project.

Considering inventory in service parts logistics network design problems with time-based service constraints

Publication No. _____

Mehmet Ferhat Candas, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Erhan Kutanoglu

We study the integrated logistics network design and inventory stocking problem as characterized by the interdependency of design and stocking decisions in service parts logistics. These two sets of decisions usually have been considered sequentially in practice, and their associated problems have been tackled separately in the research literature. The overall problem is typically further complicated due to time-based service constraints that provide lower limits for the percentages of demand satisfied within specified time windows. We introduce an optimization model that explicitly captures the interdependency between network design (location of facilities, and allocation of demands to facilities) and inventory stocking decisions (stock levels and their corresponding stochastic fill rates), and present computational results from our extensive experiments that investigate the effects of several factors including demand levels, time-based service levels, and costs. Our findings indicate that the integrated approach can provide significant cost savings over the decoupled

approach (solving the network design first and inventory stocking next), shifting the whole efficient frontier curve between cost and service level to superior regions. Furthermore, we show that the decoupled and integrated approaches may generate totally different solutions, even in the number of located facilities and in their locations, magnifying the importance of considering inventory as part of the network design models.

Our analysis consists of a special case of integrated logistics network design and inventory stocking problem in service parts logistics, where each customer requires a certain time-based service level. Introduced is a nonlinear mixed integer optimization model that is beyond our current solution technologies, yet it explicitly captures the interdependency between network design (locating facilities, and allocating customers to facilities) and inventory stocking decisions (stock levels and their corresponding stochastic fill rates). We provide two different linearized mixed integer formulations for this problem that can solve small and medium size instances. We reveal that this problem can be formulated as a capacitated facility location problem with polynomially solvable sub cases. However, it is still a challenging problem for which we have a Lagrangian-relaxation based approach that provides extremely tight lower and upper bounds.

By applying the methodology and insights of the customer-centric problem, we succeed in providing upper and lower bounds to the original system-wide service level problem, even with large instances and with medium demand levels.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Service Parts Logistics

Increasing worldwide competition and shrinking profit margins have been forcing high technology product manufacturers to differentiate themselves from others in several ways. Providing fast, high-quality after-market service is an important way to achieve this. After-market service is providing necessary service and replacement parts to existing, geographically dispersed customers when they experience any problem with their product. The service is provided as part of the contract between the customer and the manufacturer, therefore designing and operating a logistics network capable of serving customers in a time-responsive manner is crucial for after market service in which service parts logistics is a critical part.

Service parts logistics (SPL) is the set of activities that includes designing a responsive logistics network of part stocking facilities, deciding inventory ordering policies, stocking parts, and dispatching the required parts from network facilities to the customers in need, all associated with the after-market service. A major challenge in SPL is to provide the service and satisfy customer's request within the committed time. According to a survey Cohen

et al. (1997) conducted in the early 90's, after market service revenues are equal to 30% of product sales on average, and 63% of these revenues come from service maintenance contracts. The same survey finds that average operating cost in SPL is composed of inventory investment (28.6%), transportation (8.4%), warehousing (14.1%), obsolescence, scrap, and shrinkage (17.7%), and other costs (administrative, personnel and miscellaneous, 31.20%). A more recent survey by Poole (2003) shows that 40 to 50% of profits manufacturing companies come from parts, maintenance and servicing. In certain industries, customers spend 5 to 20 times the initial sale price on subsequent service and consumables. As a result of this trend, SPL has emerged as a $21 billion industry. Therefore, significant savings can be achieved by improving decision making in SPL with a goal of designing and operating an efficient logistics network and inventory system.

Locating inventory stocking facilities, allocating customer demands to these facilities and selecting stock levels maintained at these facilities are the main decisions while designing the SPL system. Traditionally, location and allocation decisions (collectively called *logistic network design* or LND) are considered part of strategic and long-term decisions, which are typically made before any tactical decisions such as inventory levels. However, redesigning an existing network more frequently is becoming more affordable due to outsourced warehousing and delivery services. For example, companies such as UPS and FedEx provide access to an extensive global logistics network and offer services ranging from sourcing of parts from vendors to shipments to

customer sites. This allows companies using third party logistics services to expand or shrink their network as needed without much difficulty. Moreover, due to the time-based service level requirements that are critical part of any SPL system, there is a stronger interaction between "strategic" network design and "tactical" inventory decisions as the service requirements are not only a coverage issue (whether a customer's demand is covered by a nearby facility), but also a function of the part availability at that facility. Thus, we conjecture that considering the effects of network decisions on inventory (and vice versa) in an integrated model becomes critical for ultimate optimization of an overall SPL system.

With this main goal in mind, we give the following example to further motivate our study: IBM, one of the largest information technology companies in the world, has a group in its Global Services Division, called Service Parts Solutions (SPS). SPS handles the after-market services for all customers of IBM (i.e., when there is a hardware problem that is under IBM's service contract, SPS is responsible for providing the necessary (service) parts to the customer along with a repair technician).

There are many different types of service parts and they perform many different functions. For example, to keep a computer system (such as a net-server) operational, the customer may need a CD-drive which is inexpensive, or a data storage unit which is expensive. Clearly, these parts differ in terms of their failure frequencies (which generate demands for service parts) and their criticality levels. The customers may be willing to wait a few days for

a CD-drive, but a few hours of delay for a storage unit may be too costly for such customers as banks or Internet-based retailers (e.g. Amazon).

Hence, the service parts can be classified in terms of their costs (low cost versus high cost), criticality (critical or not), and demand rates (high and low demand levels). In general, critical parts are made more reliable, so they have low failure rates (very infrequent demand) and they are more expensive. These parts are called "low demand high cost parts." For these parts, customers sign service level contracts requiring eight, four, and/or even two-hour response times, which are grouped under *"same day service."* For the customers that require *"rapid response,"* namely service within very short time, IBM utilizes *"outside locations"* (stocking rooms at or very near the customer sites). For the customers that require *"next day service"* (not so critical), IBM makes customers that require *"next day service"* (not so critical), IBM makes direct shipments from its central and regional distribution centers. In this study, we focus on *low demand, high cost parts* requiring *same day service*, since they play an important role in designing the network (selecting the facility locations) and finding the part stocking levels.

To illustrate the existing interactions between network design and inventory decisions, we use a small problem instance derived from the real IBM data. The data is based on a single product (for simplicity assumed to be made of a single part) currently in use in the New England states (Maine, Vermont, New Hampshire and Massachusetts). There are 26 demand points (5 digit zip codes) with average demand rates ranging from 0.49 to 1.97, with

an overall average of 0.95 units per year per demand point. The part costs about $1,200 (with 25% unit carrying charge, the annual holding cost is about $300 per unit). There are 6 candidate facility locations. We assume a fixed facility cost of $1,000 per year, and the transportation costs are based on the actual distances between customers and facilities, ranging from $40 to $500. (Figure 1.1 shows the map of the problem). We require that 70% of the total demand be satisfied within 2 hours. We assume that a customer is in the 2-hour service window of a facility if it takes less than 2 hours to serve the customer from that facility.

One can use several modeling and solution techniques to approach this problem. We list 3 alternatives here:

- Using a set covering-based model, locate the minimum number of facilities that can cover the 70% of demand. Assign each customer to an open facility with the least transportation cost. With the total demand assigned to the open facilities, find the lowest values of stocking levels such that the overall 2-hr service is guaranteed. Figure 1.2 shows the solution of the example problem with this approach. Table 1.1 lists the open locations, and summarizes the demand allocations and stock levels.

- Using a fixed-charge uncapacitated facility location model with standard distance-based demand coverage constraints, locate facilities and allocate demands to facilities. With the total demands assigned to the open facilities, find the part availabilities and lowest stocking levels guaranteeing

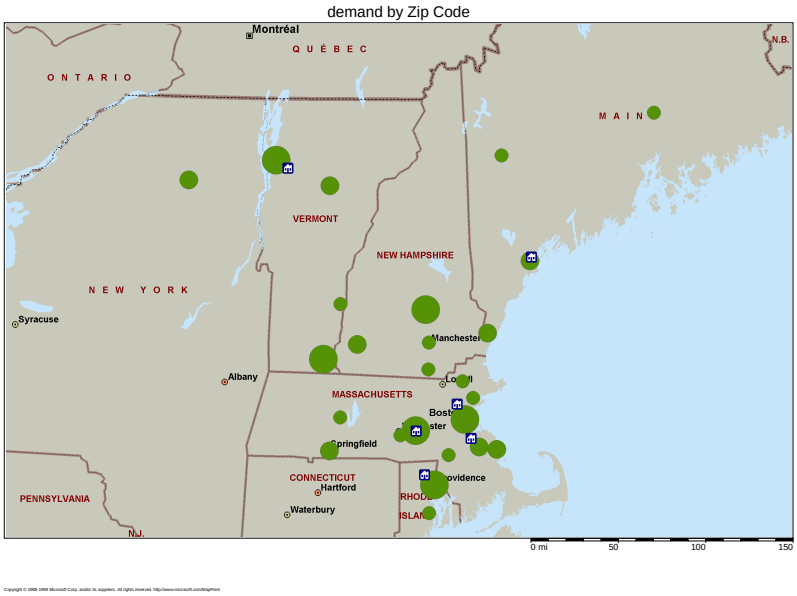


Figure 1.1: Demand points and candidate facility locations for the example instance

the target 2-hr service. Figure 1.3 shows the solution, with demand allocations and stock levels listed in Table 1.2.

- Using a fixed-charge uncapacitated facility location model with demand coverage constraints that explicitly take into account the part availabilities that depend on the stocking levels and demand allocations, locate facilities, allocate demands and find stocking levels guaranteeing the 2 hour service target. Figure 1.4 shows the map of the solution, with demand allocations and stock levels in Table 1.3.

Table 1.4 summarizes cost distributions of the three approaches. All three approaches open 3 of the candidate facilities. The set covering based

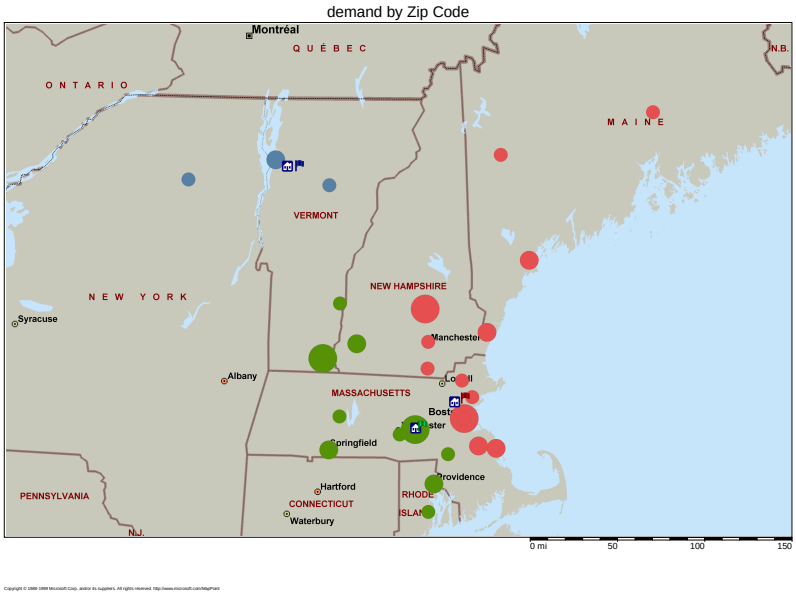


Figure 1.2: Map of the solution using Set covering-based approach

Table 1.1: Open facility locations, demand allocations and stock levels (Set covering based approach)

Open Facility	Total Assigned Demand	Stock level
Green (01581)	9.894340	2
Red (01801)	11.378464	2
Blue (05495)	3.463010	1

Table 1.2: Open facility locations, demand allocations and stock levels (UFL based approach)

Open Facility	Total Assigned Demand	Stock level
Green (01581)	18.304510	3
Orange (04103)	2.968294	1
Blue (05495)	3.463010	2

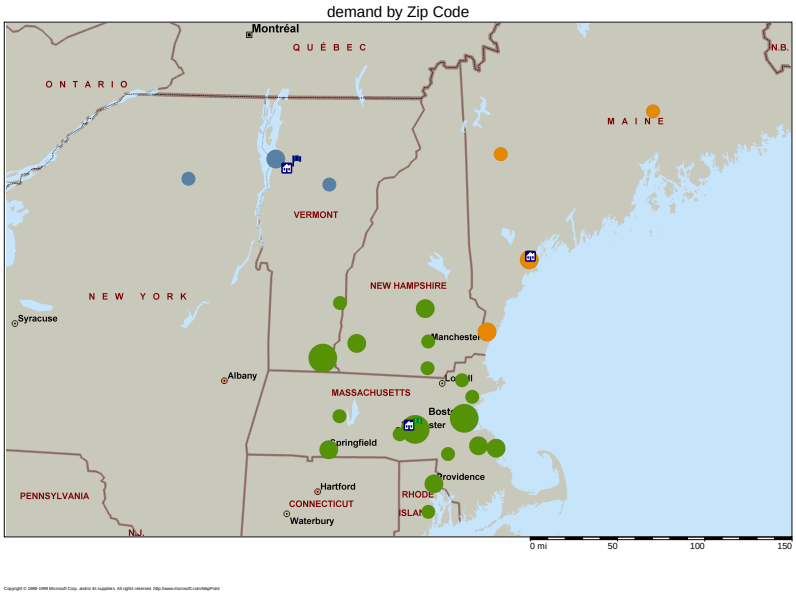


Figure 1.3: Map of the solution using UFL-based approach

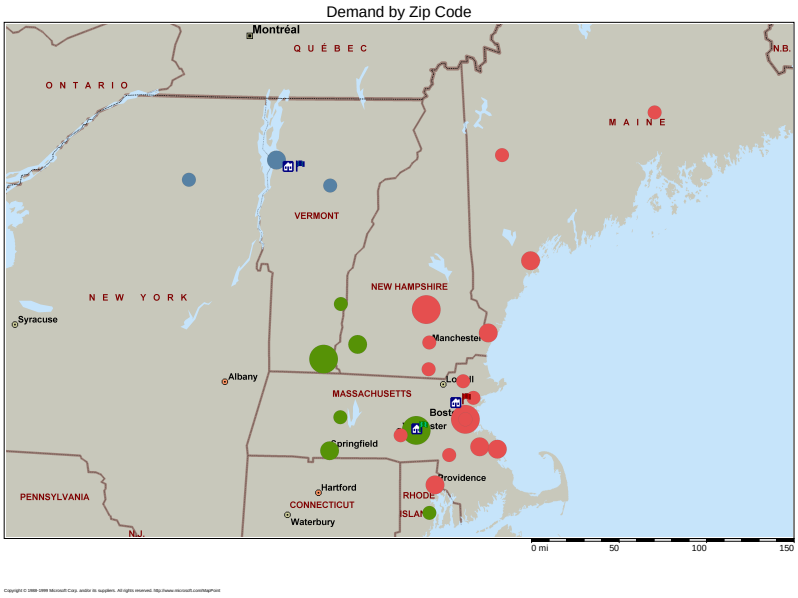


Figure 1.4: Map of the solution using Integrated approach

Table 1.3: Open facility locations, demand allocations and stock levels (Integrated approach)

Open Facility	Total Assigned Demand	Stock level
Green (01581)	7.420766	1
Red (01801)	13.852038	2
Blue (05495)	3.463010	1

approach and the integrated approach open the same 3 locations. The main difference between their solutions is due to demand allocations, which in turn affects their transportation costs and ultimately their stock levels and associated inventory costs. The UFL-based approach saves some transportation costs, but, due to the open facilities and demand allocations made without any regard to inventory costs, its inventory cost is high. The integrated approach provides the lowest total costs, as it combines all relevant costs and decisions in a single model.

Note that the UFL-based approach has higher total costs than that of the set-covering based approach for this example. This may seem counter intuitive at first since UFL-based approach takes more of the available information into account and integrates location and demand allocation decisions, but there is no guarantee that it will improve the total costs as the total includes inventory costs.

Table 1.4: Cost comparisons for the three approaches

	Set-covering based	UFL-based	Integrated
Location cost	3,000	3,000	3,000
Transportation cost	2,550	2,515	2,615
Inventory cost	1,518	1,822	1,214
TOTAL	7,068	7,337	6,829

1.2 Literature Review

In this study, we investigate the benefits of explicitly considering inventory within a logistics network design setting by integrating two traditionally separate problems: network design, and inventory management. Therefore, there are related papers from (1) location/allocation/network design literature, (2) multi-location service constrained inventory management, and, of course, (3) integration issues. We now briefly review each of these areas.

1.2.1 Facility location/network design problems

Facility location and network design problems with countless variations have been studied extensively in the literature. The relevant papers in this area study service constrained, stochastic, or reliability-based problems. There are studies that address global uncertainties such as exchange rates, transfer prices, taxes and market prices along with supplier reliability and lead time uncertainty in a single-echelon model (Vidal and Goetschalckx, 2000), and in a multi-echelon model (Bundschuh et al., 2003). These studies define reliability as the probability of being on time or as the amount of product shipped in

comparison with a specified target value.

Snyder and Daskin (2004) investigate the uncapacitated facility location (UFL) problem with potentially unreliable facilities. In this reliability-based model, a customer may not be served from its facility due to "failure" of the facility, which occurs with a certain probability. When this happens, the customer is served by one of a series of facilities, incurring a higher transportation cost. (In our model, we can view fill rates as facility reliability measures, but we have multiple fill rates (one calculated for each part) at a facility and these fill rates depend on both the stock levels and the amount of demand assigned to the facility, both of which are decided as part of the solution).

Two recent reviews (Daskin et al., 2003; Snyder, 2004) in this area surveys these papers, focusing on more recent accomplishments on solving extended facility location models in the supply chain management context. Daskin (1995) is a reference text on discrete facility location problems. Magnanti and Wong (1984) review the early literature on the facility location problems, while Drezner (1995) summarizes the overall research effort by 1995.

1.2.2 Inventory Management and Service Parts Logistics

Similar to the location literature, there is a vast amount of work on inventory management. We only review the ones that are most related to our work and refer the reader to Zipkin (2000) for a recent text on the topic. A stream of research related to the inventory-portion of our study explicitly considers service level constraints. We can list Chen and Krass (2001) for

a single facility operating reorder point and order-up-to level policy (s, S), and Agrawal and Seshadri (2000) for order-quantity and reorder point policy (Q, r).

A limited number of papers investigate the multi-item setting in which an overall fill rate for a group of items ("order") is targeted through adjusting item stock levels, which is a form of "service allocation." One example (Song, 1998) also investigates a simplified time-based service level for base stock policies.

One of the few papers studying multi-facility problems is by Rappold and Muckstadt (2000) who propose an algorithm to allocate inventory across facilities operating base stock policies in a multi-echelon network structure.

Early literature on spare/service parts management in multi-echelon systems includes (Sherbrooke, 1968, 1986), and (Muckstadt, 1973). Successful applications in service parts logistics systems include such industries as automotive (Cohen et al., 2000), computer and other electronic equipment service (Cohen et al., 1988, 1990, 1999), and military (Rustenburg et al., 2001). One of the early works on multi-echelon service parts inventory management is by Muckstadt and Thomas (1973). A limited number of studies in this group consider fill rate-based service allocation (Cohen et al., 1988, 1989, 1992),

1.2.3 Network Design Models with Inventory Considerations

Integration of facility location and inventory problems is a very recent research area, hence there are only a handful of papers on this topic. As they

are most relevant to our study, we review them in more detail.

The most relevant work is done by Jeet and Kutanoglu (2005), who attack the overall integrated problem directly, considering the part commonality issues in a multi-product setting, using a fill rate approximation scheme in the process. One of the early papers modifies the uncapacitated facility location problem to implicitly consider limited inventory levels (Barahona and Jensen, 1998). Nozick and Turnquist (1998) approximate inventory costs as part of the fixed facility costs and propose a model that maximizes service coverage. This is extended to minimize costs subject to service coverage constraints in Nozick (2001). The paper by Daskin et al. (2002) is probably the first study that explicitly includes inventory costs as part of a simple, uncapacitated facility location model. Their model assumes economic order quantity (EOQ)-based ordering and constant fill rate-based safety stocks across all facilities. The total cost function including the inventory related terms makes the overall model a nonlinear integer program which is then solved using Lagrangian relaxation. Shen et al. (2003) develop a column generation-based method to solve the same model. Ozsen et al. (2004) recently extend the model to include facility capacities. A parallel and a very similar model is analyzed in (Miranda and Garrido, 2004).

Other related papers include a grid-based location-inventory model (Erlebacher and Meller, 2002), a scenario-based stochastic extension of Daskin et al.'s (2002) joint location-inventory model (Snyder et al., 2003), an approximation algorithm for the joint location-inventory model that ignores trans-

portation costs (Teo et al., 2001), and finally the same model extended with customer service levels (Shen and Daskin, 2003).

1.3 Contributions

In the light of this literature review, our study has three main contributions. First, we consider inventory decisions (stock levels) and their costs explicitly in a multi-part network design model, thus making service levels (part availabilities or fill rates) vary across facilities and parts to achieve a system-wide time-based service level. To the best of our knowledge, this is done for the first time. In particular, service allocation across facilities in an integrated multi-part network design and inventory model is new. Second, we explicitly compare the proposed integrated model with the conventional approach of making network design decisions first followed by making stocking decisions (the decoupled approach). The papers above develop solution techniques to simpler versions of our integrated model, pre-assuming a significant benefit from integration. Here, we make this comparison between the decoupled approach and the integrated approach explicitly by quantifying these benefits and showing the conditions under which the improvement due to integration is significant, and those under which the decoupled approach is just good enough. Finally, we develop solution methodologies for different versions of the integrated problem, such as the system-wide service level and the customer-centric service level problems.

1.4 Outline

This dissertation contains six chapters. In the next chapter, we propose an integrated approach for the system-wide service level problem. By comparing it with traditional approaches we show the benefits of considering inventory decisions together with network design decisions. In chapter three, we analyze a special case of integrated logistics network design and an inventory stocking problem in which each customer requires a certain time-based service level. This problem is different from the system-wide problem we propose in the second chapter in terms of having a separate service level constraint for each customer instead of having a single combined service level constraint for overall system. With the insights we gained from the customer-centric problem, we revisit the system-wide service level problem in chapter four and present the results. In chapter five we discuss the extensions and in chapter six we present our final conclusion, discuss the implications of our findings and identify possibilities for future research.

Chapter 2

System-Wide Service Levels

2.1 Introduction

Motivated by the real challenges in today's SPL systems, we model the integrated network design and inventory stocking problem. We explicitly consider inventory decisions and costs in the LND problem, which is itself already complicated due to explicit time-based service constraints. We also quantify the benefit of considering both network design and inventory decisions in the same model and identify the conditions and problem settings in which this benefit is significant.

We make the following assumptions to facilitate model development:

- We assume that network design involves stocking facilities that are all in one echelon facing the direct demand from geographically dispersed customers. We assume that these facilities to be located are replenished from a central warehouse with infinite capacity (that is, the central warehouse can replenish the stocking facilities anytime without any delay). The lead times from the central warehouse to all facilities is known and constant.

- Due to the low-demand nature of the motivating SPL problem, we assume that the facilities use a continuous review, one-for-one (or base-stock, also called $(S-1, S)$) replenishment policy. This is typical as demands are low, and lead times are relatively short in SPL systems; hence, there is no incentive to order in batches, especially considering the very low ordering costs due to bundled outsourced transportation services. This is a very common assumption in the low-demand item inventory literature. (For example, all the models in the text by (Sherbrooke, 1992) use this policy, even for higher-echelon facilities, where demands from lower-echelon facilities are aggregated.)

- We assume that demand for each part at each demand point arrives one at a time according to an independent Poisson process, which is typical in low-demand settings. We further assume that we know the mean demand rates obtained from the part failure rate distributions and the number of parts used at each demand point. The Poisson assumption is again extremely common in the SPL literature (see, e.g., Sherbrooke (1992); Muckstadt (2005)). (Although the model developed here can be written for any unit of time, in the following discussion we assume that demands and costs are annual for ease of illustration). Any unsatisfied demand due to a stockout at a facility is backordered. Note that demands (failure rates) for high-cost, critical parts in SPL systems are very low, as these parts are made extremely reliable.

- We assume that demand points represent aggregation of individual actual

customer locations (e.g., a 5-digit zip code could be a demand point representing all the real customers in that zip code). Moreover, we assume that all the individual customer service level requirements are aggregated to obtain a target service level for each part in the region. This is one way to deal with the scale of the individual customer service contract requirements. The aggregate target service level for the region can be a demand-weighted average of all individual customer requirements. An alternative can be choosing the highest level of all customer requirements as the service level for the region. We assume that service contracts, demand aggregation across individual customers to form demand points, and service level aggregation to obtain a target service level for the region are done a priori. This translates into a percentage of demand to be satisfied within a certain service time window. For example, a typical aggregate service level may read "70% of total demand for part 1 must be satisfied from facilities that are within 4 hours of the demand points."

- We assume that one service time window is defined. Although typical SPL systems have multiple "tiered" service time windows (with increasing service level requirements for longer time windows), usually one of the time windows is the most restrictive, hence assuming one window for each part should not hinder the model's value. Besides, modifying the model (as will be seen) for multiple windows is straightforward. In the experiments, however, we vary the time window as a control factor to see its effect on the results.

- We assume that we know which customers a facility can serve within the service time window. As this is usually a function of distance and the mode of transportation available to the facility and customer, we assume that this processing of transportation times is performed for each customer and facility pair a priori. We further assume that each customer's part request is satisfied by a single direct shipment from a facility, without any shipment consolidation or bundling. Not only this is an actual practice in SPL systems, but also it is very unlikely that there is the time or opportunity to consolidate multiple shipments due to low demand and strict time windows.

- We finally assume that it is possible to split a part's demand at a demand point while allocating it to facilities , i.e., demand can be satisfied by more than one facility. Due to the structure of the problem, this does not happen very often in optimal solutions, but it provides opportunities for the model to find superior inventory stocking level combinations by adjusting the total demand assigned to facilities through splitting demand. From this perspective, the partial demand allocated to a facility can be viewed as the long-run fraction of the demand to be satisfied from the facility. (Modeling the no-split assumption is equally easy, without much potential to change our overall findings and insights.) At the operational level, this means assigning demand to facilities randomly with probabilities corresponding to their long-run fractions found in the optimal solution. We assume that the facility satisfies the demands in the

order of their arrival (i.e., first come first served) regardless of where the requesting demand point is (regardless of whether the demand point is within the time window). With this, total demand assigned to each facility follows the Poisson distribution with the corresponding aggregate mean demand. We assume that there is no inventory transfer or transshipment between stocking facilities. Although some SPL systems use transshipments in case of stockouts to improve service and pool risk, calculating stock levels, even for a given fixed facility location with known demand, is a nontrivial task (Cohen et al. (1986); Alfredsson and Verrijdt (1999)). As we show here, the new integrated network design and inventory model without transshipments is rich enough to lead to interesting results. In that sense, this chapter serves as a precursor to study more complex models with transshipments.

In section 2.2 we define the problem and introduce the model for the integrated approach. This is followed by the computational results for this approach in Section 2.3. Section 2.4 details a UFL-based approach called *decoupled approach* to show the benefits of integration. We compare the two approaches (decoupled and integrated) in Section 2.5.

2.2 Problem Definition and Modeling

We first define the problem setting and introduce the notation, and then introduce the integrated model formulation, dealing with its complexities along the way.

2.2.1 Problem Setting and Notation

For a given set of customers and their demands, we seek to (1) locate a set of stocking facilities selected from a set of candidate facility locations, (2) allocate customer demands to these located (open) facilities, and (3) determine stock levels to be used at the open facilities. Decision sets (1) and (2) make up the *network design*, and (3) are the *inventory stocking* decisions. The goal is to make these decisions with minimum possible total facility, transportation, and inventory costs while achieving the target (required) service levels.

We now introduce the notation. We are given a set of candidate facility locations I (indexed by i), a set of demand points J (indexed by j), and a set of parts K (indexed by k). When we open facility i (or more correctly, locate a facility at candidate location i), we incur an annual cost of f_i for operating the facility. The unit transportation cost between facility i and demand point j for part k is c_{ijk}. Let τ_{ij} be the transportation time from facility i to demand point j. Comparing τ_{ij} to the service time window w, we obtain δ_{ij}, the identifier which takes value 1 if facility i can ship a part requested at demand point j within the specified service time window ($\tau_{ij} \leq w$), 0 otherwise ($\tau_{ij} > w$). The mean annual demand rate is d_{jk} for part k at demand point j (i.e., the rate at which demand point j experiences part k failures). With a little notation abuse, we denote the total annual mean demand rate for part k (across all customer demand points in the region) by d_k, $d_k = \sum_{j \in J} d_{jk}$. The target level for the system-wide service for the specified service time window is α_k for part k. Hence, $100\alpha_k\%$ of the total annual demand for part k (of all part k failures)

must be satisfied within the time window.

We now develop the integrated model, which simultaneously makes network design and inventory stocking decisions and captures the true relationship among location, demand allocation, fill rate-based part availability, and system-wide service levels. Using a classical facility location formulation as a base, we first build an intuitive, but highly nonlinear integer programming model (Section 2.2.2). We then show how one can take advantage of the low-demand rates (hence, low stock levels) to linearize the model (Section 2.2.3). We finally add a post-processing stage to eliminate the drawbacks of the linearization and to refine the model's solution (Section 2.2.4).

2.2.2 Nonlinear Integer Programming Model

Traditional network design models are based on modifications of the (fixed-charge, multi-commodity) uncapacitated facility location problem, with the main change being the addition of the service coverage constraints. We now introduce this classical model and then show how we further modify it to obtain the integrated model, which captures the network design-inventory interaction.

The model uses the following decision variables:

- X_{ijk} = long run fraction of part k's demand at demand point j allocated to facility i
- $Y_i = 1$ if facility i is open, 0 otherwise

Using allocation variables X_{ijk}, we define the total demand rate at facility i for part k as follows:

$$d_{ik} = \sum_{j \in J} d_{jk} X_{ijk}. \tag{2.1}$$

The demand rate at which facility i experiences part k failures from customers within the time window is

$$d_{ik}^{\leq w} = \sum_{j \in J} \delta_{ij} d_{jk} X_{ijk}. \tag{2.2}$$

Hence, the fraction of part k failures from customers within the time window for facility i is $d_{ik}^{\leq w}/d_{ik}$. Moreover, the fraction of part k failures that facility i serves is d_{ik}/d_k out of all regional demand for part k. The service coverage constraints state that the fraction of all failures that can be addressed from facilities within the time window is at least α_k for part k. We then write the service coverage constraints as follows:

$$\sum_{i \in I} \frac{d_{ik}^{\leq w}}{d_{ik}} \frac{d_{ik}}{d_k} \geq \alpha_k. \tag{2.3}$$

The multi-part, service-coverage constrained, uncapacitated facility location model is as follows:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} d_{jk} X_{ijk} \tag{2.4}$$

$$\sum_{i \in I} X_{ijk} = 1, \quad \forall j \in J, k \in K \tag{2.5}$$

$$X_{ijk} \leq Y_i, \quad \forall i \in I, j \in J, k \in K \tag{2.6}$$

$$\sum_{i \in I} \sum_{j \in J} \delta_{ij} \frac{d_{jk}}{d_k} X_{ijk} \geq \alpha_k, \ \forall k \in K \tag{2.7}$$

$$0 \leq X_{ijk} \leq 1, \ \forall i \in I, j \in J, k \in K \tag{2.8}$$

$$Y_i = 0 \text{ or } 1, \ \forall i \in I. \tag{2.9}$$

The objective (2.4) is to minimize the total expected fixed facility opening costs and transportation costs. Constraints (2.5) guarantee that all demands at all demand points will be fully satisfied. Constraints (2.6) mean that any facility serving a demand point should be open. As rewritten versions of (2.3) explicitly including allocation decision variables X_{ijk}, constraints (2.7) ensure that the required percentage of the total demand for each part is *assigned* to a facility within the specified time window of the demand points.

The implicit assumption made with constraint (2.7) is that the assignment of demands to facilities within the service time window guarantees delivering the part within the window. That is, a facility is capable of contributing towards system-wide service level for the demands that can be reached within the service time window from it. (In fact, this formulation becomes the first-stage submodel of the decoupled approach, which is introduced later). However, the actual service is not only a function of the transportation time (or distance) between customers and their assigned facilities. The actual service is also a function of part availability: If a facility does not have the required part in stock when needed, having the facility close, even next door, to its requesting customer will be useless. More interestingly, the part availability at a

facility is a function of the demand assigned to it, which is just being decided by variables X_{ijk} for all i, j, k as part of the model: For a given stock level at the facility, as we assign more demand to a facility, its part availability deteriorates. Part availability is usually captured as the long run fill rate, which is defined as the long run fraction of demand satisfied directly from stock on hand.

We now define the mean lead time demand at facility i for part k:

$$\lambda_{ik} = t_{ik} d_{ik} = t_{ik} \sum_{j \in J} d_{jk} X_{ijk} \tag{2.10}$$

where t_{ik} is the replenishment lead time for part k at facility i, measured in years. Due to the assumptions we made earlier, the lead time demand for part k at facility i is Poisson with mean rate λ_{ik}. Also, let S_{ik} be the base stock level for part k at facility i. We can now define $\beta_{ik}(S_{ik}, \lambda_{ik})$ as the fill rate for part k at facility i with (base) stock level S_{ik} and mean lead time demand λ_{ik}. As demands (part failures) occur one at a time according to a Poisson process and the demands that cannot be filled immediately from stock are backordered, from the PASTA (Poisson Arrivals See Time Averages) property, the fill rate is equal to the limiting probability of having stock on hand (Zipkin, 2000; Muckstadt, 2005). Then, for a given mean lead time demand λ_{ik} and stock level S_{ik} for part k at facility i, the fill rate is

$$\beta_{ik}(S_{ik}, \lambda_{ik}) = G_{ik}(S_{ik} - 1) = \sum_{r=0}^{S_{ik}-1} \lambda_{ik}^r e^{-\lambda_{ik}} / r! \tag{2.11}$$

where $G_{ik}(S_{ik}-1)$ is the cumulative (Poisson) distribution function of the lead time demand for part k at facility i evaluated at $S_{ik} - 1$. For the reasoning

behind this fill rate formula, consider a specific replenishment order for part k at facility i. Since the lead times are assumed to be constant (say, one week), we know that all the other $S_{ik} - 1$ items either in inventory or on order will be available to fill new demand before the order under consideration arrives. Therefore, the only way the order can arrive after the demand for it has occurred is if demand during the lead time is greater than or equal to S_{ik}. Then, the probability that the order arrives before its demand is given by the probability that the lead time demand is smaller than S_{ik}, or $G(S_{ik}-1)$. Since all orders are alike with regard to this calculation, the fraction of demand that is filled from stock is equal to the probability that an order arrives before the demand for it has occurred (Hopp and Spearman (2000)).

Note that the dependency of fill rate and lead time demand (which in turn is a function of demand allocation decisions, X_{ijk}'s) is implicit in the cumulative distribution function. An illustration of fill rate as a function of the mean lead time demand and the stock level is shown in Figure 2.1 (depicted for two stock levels).

With the fill rates calculated correctly, the more accurate service levels that include the possibility that the requested parts may not be available at the facilities when needed can now be written as follows:

$$\sum_{i \in I} \frac{d_{ik}^{\leq w}}{d_{ik}} \frac{d_{ik}}{d_k} \beta_{ik}(S_{ik}, \lambda_{ik}) \geq \alpha_k, \tag{2.12}$$

or in terms of the original demand allocation decision variables:

$$\sum_{i \in I} \sum_{j \in J} \frac{\delta_{ij} d_{jk} X_{ijk}}{d_k} \beta_{ik}(S_{ik}, \lambda_{ik}) \geq \alpha_k. \tag{2.13}$$

Note that this constraint is similar in spirit to the service level constraints developed in (Muckstadt, 2005) for a given set of stocking facilities and their already allocated Poisson demands. In this new form, which is needed in an integrated network design and stocking framework, the constraint is much more complex (and nonlinear), but is indeed critical for capturing the correct relationships among all elements of the overall system; facilities, transportation, inventory, and service. For example, if the demand assigned to a facility generates a mean lead time demand of 0.1, the fill rate at that facility for a stock level of 1 is about 90%. The fill rate goes up with increasing stock levels but this comes at the expense of increasing inventory costs. Therefore, we add inventory costs to the objective function to capture all the trade-offs among facility, transportation, and inventory costs. In this way, the new model captures the interactions between these decisions and service.

We now write the complete model. To facilitate development, we assume that there is a finite number of alternatives for stock levels S_{ik}. We denote this set of all possible stock levels by L, and denote its largest element by L. Hence, we consider $S_{ik} \in \{0, 1, 2, \ldots, L\}$ for all i, k. For example, typical stock levels at low-echelon facilities in most SPL systems are usually 0, 1, sometimes 2, and very rarely 3. As SPL systems deal with extremely low demand rates, considering higher stock levels is unnecessary and treating stock levels as being continuous is not an option. As the main idea here is to capture the fill rates for changing stock levels and compute their inventory costs, we make this set large enough to give practically 100% fill rate at the

largest stock level L for a conservative estimate of total demand that could be assigned to a facility. In this case, even if we included larger stock levels in the set, the model would not consider them as any additional unit would increase inventory costs without providing additional improvement in fill rate and service. Finally, let h_{ik} be the unit inventory holding cost of part k at facility i. The overall model is as follows:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} d_{jk} X_{ijk} + \sum_{i \in I} \sum_{k \in K} h_{ik} S_{ik} \tag{2.14}$$

$$\sum_{i \in I} X_{ijk} = 1, \ \forall j, k \tag{2.15}$$

$$X_{ijk} \leq Y_i \ \forall i, j, k \tag{2.16}$$

$$S_{ik} \leq L Y_i, \ \forall i, k \tag{2.17}$$

$$\sum_{i \in I} \sum_{j \in J} \frac{\delta_{ij} d_{jk} X_{ijk}}{d_k} \beta_{ik}(S_{ik}, \lambda_{ik}) \geq \alpha_k, \ \forall k \tag{2.18}$$

$$\lambda_{ik} = t_{ik} \sum_{j \in J} d_{jk} X_{ijk}, \ \forall i, k \tag{2.19}$$

$$\beta_{ik}(S_{ik}, \lambda_{ik}) = \sum_{r=0}^{S_{ik}-1} \lambda_{ik}^r \frac{e^{-\lambda_{ik}}}{r!}, \ \forall i, k \tag{2.20}$$

$$0 \leq X_{ijk} \leq 1, \ \forall i \in I, j \in J, k \in K \tag{2.21}$$

$$Y_i = 0 \text{ or } 1, \ \forall i \in I \tag{2.22}$$

$$S_{ik} \in \{0, 1, 2, \ldots, L\}, \ \forall i, k. \tag{2.23}$$

The new term in the objective (2.14) is the total inventory investment. Constraints (2.17) allow stock levels to be greater than 0 only for open facilities. Constraints (2.18) are the time-based service coverage constraints, in which the mean lead time demand rates are calculated in constraints (2.19) and the

fill rates in constraints (2.20). Finally, constraints (2.23) allow the stock levels to be selected from the initially developed set of integer stock levels.

This model is clearly a nonlinear, mixed integer programming problem. The service level constraints (2.18) are the main source of nonlinearity, since they combine demand allocation variables X_{ijk}'s with the fill rate variables $\beta_{ik}(S_{ik}, \lambda_{ik})$, which are themselves nonlinear functions of two other decision variables, stock levels S_{ik} and mean lead time demands λ_{ik}, which are, in turn, functions of demand allocation variables X_{ijk}'s. (This circular relationship among location, demand allocation, stock level, and then service provides yet additional evidence that a comprehensive model that captures them all is needed). To be able to solve this problem with available optimization techniques, we propose to linearize the service level constraints. This process is the topic of the next subsection.

2.2.3 Linearized Model

We first approximate the nonlinear fill rate function with a step function. The motivation behind this development is the possibility that we can tabulate potential fill rates for a set of demand levels and stock levels a priori, so that complex fill rate and mean lead time calculations can be done by table-lookups. We then separate the nonlinearity using additional variables and finally remove the nonlinearity in the service level constraints. The idea here is to take advantage of the 0-1 nature of the variables used for table-lookups.

As shown in Figure 2.1 for a typical service part, the fill rate is a decreasing function of the mean lead time demand for a given stock level. For each potential stock level, we seek to approximate the fill rate with a step function. Figure 2.1 shows step functions to be used to approximate for the corresponding (continuous) fill rate functions for two selected stock levels. Here, we take advantage of having low-demand parts. We need to consider only a few potential stock levels, say at most up to 5. As the range of mean lead time demands is also small, we can use a small number of "demand intervals" (steps in the fill rate step function, each corresponding to an interval on the demand axis) to compute approximate fill rates. A tabulated version of the step function calculations for two stock levels (1 and 2) is given in Table 2.1.

We define the following additional notation to detail our approximation. Suppose we divide the mean lead time demand axis into N intervals. The intervals are indexed by n and the right end point of each interval is denoted by a_{kn} (these are break points on the demand axis for each part, where $a_{k0} = 0$). Hence, when the mean lead time demand falls in the interval $(a_{k,n-1},\ a_{kn}]$, its fill rate for a stock level is approximated as the (actual) fill rate of the mid-point of the interval evaluated at the same stock level. More specifically, let b_{kln} be the approximated fill rate when the mean lead time demand is in $(a_{k,n-1},\ a_{kn}]$ and when the stock level under consideration is l. Then,

$$b_{kln} = G(l-1) \tag{2.24}$$

where $G(.)$ now is the Poisson cumulative distribution function with the mean lead time demand of $(a_{k,n-1} + a_{kn})/2$. The use of mid-point mean lead time

demand aims to prevent constant over- or under-estimation of the true fill rate. (Later, we outline a post-processing phase that will compute actual fill rates and revise stock levels when necessary.) Finally, we do a similar approximation for all potential stock levels, $l = 1, 2, \ldots, L$.

We note that the demand intervals are not necessarily of equal size; in fact we intentionally make the intervals larger as the mean lead time demand increases (as illustrated in Figure 2.1). The idea is to capture the most sensitive and useful parts of the fill rate function more accurately (with a higher resolution) in the parts of the demand most relevant to our calculations. Due to low demand levels, we are more likely to use the left-tail of the fill rate function than the approximations on the right-tail (larger demand values), which may not be used at all.

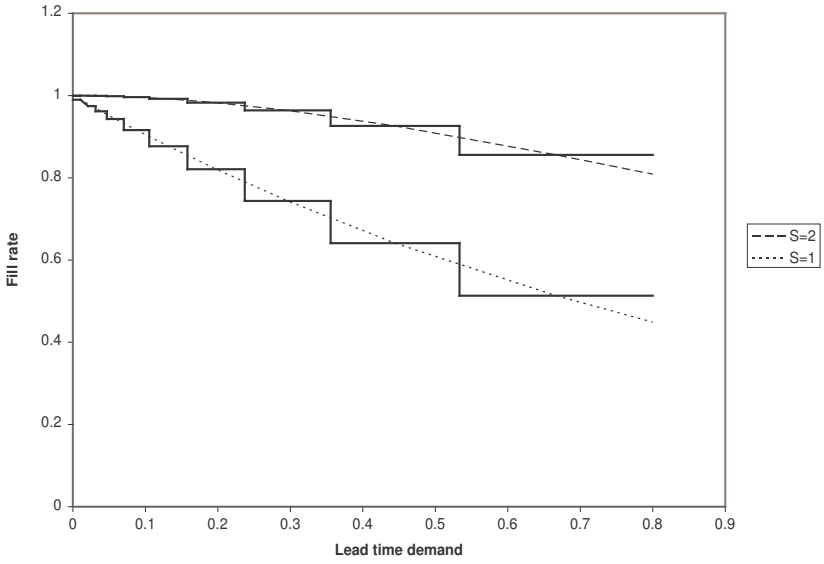


Figure 2.1: Fill rate linearization, shown for two stock levels

Table 2.1: A sample table for step function approximation of fill rates (for two stock levels, 1 and 2)

n	$(a_{k,n-1}, a_{kn}]$	b_{k1n}	b_{k2n}
0	0	1	1
1	$(0, 0.0208]$	0.9896	0.999946
2	$(0.0208, 0.0312]$	0.9743	0.999667
3	$(0.0312, 0.0468]$	0.9617	0.999258
4	$(0.0468, 0.0702]$	0.9431	0.998353
5	$(0.0702, 0.1053]$	0.9159	0.996365
6	$(0.1053, 0.158]$	0.8766	0.992054
7	$(0.158, 0.237]$	0.8207	0.982879
8	$(0.237, 0.3555]$	0.7435	0.963883
9	$(0.3555, 0.5333]$	0.6411	0.926149
10	$(0.5333, 0.8]$	0.5134	0.855695

We now formulate the table-lookup process, finding the correct demand interval in which the mean lead time demand falls, and making this process a part of the model. For this purpose, we define another binary decision variable, Q_{ikn}, which takes value 1 when $a_{k,n-1} < \lambda_{ik} \leq a_{kn}$, and takes the value 0 otherwise. Using Table 2.1, for example, if the mean lead time demand for part k at facility i is in a certain range, say $(0.0208, 0.0312]$ (for which $n = 2$), then $Q_{ik2} = 1$, and $Q_{ikn} = 0$ for all other n, $n = 1, 3, \ldots, N = 10$. We write the constraints that guarantee these assignments with the help of another set of binary decision variables, R_{ikn}, defined for all i, k, n.

$$a_{kn} - \lambda_{ik} \geq M_1(R_{ikn} - 1), \ \forall i, k, n \tag{2.25}$$

$$a_{kn} - \lambda_{ik} \leq M_2 R_{ikn}, \ \forall i, k, n \tag{2.26}$$

$$Q_{ikn} = R_{i,k,n} - R_{i,k,n-1},\ \forall i, k, n \tag{2.27}$$

$$Q_{ikn} = 0 \text{ or } 1, R_{ikn} = 0 \text{ or } 1,\ \forall i, k, n \tag{2.28}$$

$$R_{ik0} = 0,\ \forall i, k \tag{2.29}$$

where M_1 and M_2 are big-M's that can be set to their tightest values, depending on the facility, part, and demand interval to which they relate: $M_1 = t_{ik}d_k$ (the largest possible value of the mean lead time demand for part k at facility i) and $M_2 = a_{kn}$.

Here $R_{ikn} = 0$ for n values, up to the interval just before the demand interval in which the mean lead time demand λ_{ik} falls. For this and other intervals $R_{ikn} = 1$. Hence, $Q_{ikn} = 1$ for the interval that contains the mean lead time demand. In this interval, $R_{i,k,n-1} = 0$ and $R_{ikn} = 1$.

We now write the following for the approximate fill rate for given stock level $S_{ik} = l$:

$$\beta_{ik}(S_{ik} = l, \lambda_{ik}) = \sum_{n \in N} b_{kln} Q_{ikn},\ \forall i, k \tag{2.30}$$

where N is the set of numbers from 0 to the number of intervals used in the fill rate approximation ($N = \{0, 1, \ldots, 10\}$ in the example above).

To make further linearization possible for a generic S_{ik}, we define a new set of binary decision variables: W_{ikl} takes value 1 when facility i uses stock level l for part k, and 0 otherwise. With this, we write the approximate fill rates as follows:

$$\beta_{ik}(S_{ik}, \lambda_{ik}) = \sum_{l \in L} \sum_{n \in N} b_{kln} W_{ikl} Q_{ikn},\ \forall i, k \tag{2.31}$$

where W_{ikl} will be forced to take value of 1 for $l = S_{ik}$, 0 otherwise. Note that this final fill rate approximation involves multipling two binary variables (W_{ikl} and Q_{ikn}). We can plug in these versions of approximate fill rates into the service level constraints (2.18) to obtain the following (while eliminating constraints (2.20)).

$$\sum_{i\in I}\sum_{j\in J}\frac{\delta_{ij}d_{jk}X_{ijk}}{d_k}\left(\sum_{l\in L}\sum_{n\in N}b_{kln}W_{ikl}Q_{ikn}\right)\geq\alpha_k,\ \forall k\in K. \tag{2.32}$$

Constraint (2.32) for part k simply states that for "selected stock level l" (hence $W_{ikl}=1$), for the "corresponding mean lead time demand" ($Q_{ikn}=1$), and for the approximated fill rates for each facility i, the sum of "fractions of demands that are directly satisfied from facilities that are within the time window" ($\sum_{j\in J}\frac{\delta_{ij}d_{jk}X_{ijk}}{d_k}b_{kln}$) should be at least the "target service level" (α_k).

To finalize our linearization, we define a new continuous decision variable, V_{ikln} which takes the value $\sum_{j\in J}\frac{\delta_{ij}d_{jk}X_{ijk}}{d_k}b_{kln}$ when $W_{ikl}=Q_{ikn}=1$, and 0 otherwise. Hence, the service level satisfied at facility i for part k is given by $\sum_{l\in L}\sum_{n\in N}V_{ikln}$. To achieve this, we introduce the following constraints:

$$V_{ikln}\leq M_3Q_{ikn},\ \forall i,k,l,n \tag{2.33}$$

$$V_{ikln}\leq M_3W_{ikl},\ \forall i,k,l,n \tag{2.34}$$

$$V_{ikln}\leq\sum_{j\in J}\frac{\delta_{ij}d_{jk}X_{ijk}}{d_k}b_{kln},\ \forall i,k,l,n \tag{2.35}$$

$$V_{ikln}\geq\sum_{j\in J}\frac{\delta_{ij}d_{jk}X_{ijk}}{d_k}b_{kln}-M_3(1-Q_{ikn})-M_3(1-W_{ikl}),\ \forall i,k,l,n \tag{2.36}$$

$$V_{ikln}\geq 0,\ \forall i,k,l,n \tag{2.37}$$

where M_3 is another big-M with the tightest value that can be customized for each i, k, l, and n: $M_3 = \sum_{j \in J} \frac{\delta_{ij} d_{jk}}{d_k} b_{kln}$. Here, if one or both of W_{ikl} and Q_{ikn} is 0 for some i, k, l and n, then constraints (2.33, 2.34, 2.37) force V_{ikln} to be 0. For part k at facility i, if both W_{ikl} and Q_{ikn} are 1 for some l and n, i.e, say $\hat{l}$ and $\hat{n}$ ($W_{ik\hat{l}} = 1$ and $Q_{ik\hat{n}} = 1$), constraints (2.35) and (2.36) will force $V_{ik\hat{l}\hat{n}}$ to be equal to $\sum_{j \in J} \frac{\delta_{ij} d_{jk} X_{ijk}}{d_k} b_{k\hat{l}\hat{n}}$. We can now rewrite the service level constraints (2.32) as follows:

$$\sum_{i \in I} \sum_{l \in L} \sum_{n \in N} V_{ikln} \geq \alpha_k, \ \forall k \tag{2.38}$$

The following is the complete model:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} d_{jk} X_{ijk} + \sum_{i \in I} \sum_{k \in K} \sum_{l \in L} l h_{ik} W_{ikl} \tag{2.39}$$

$$\sum_{i \in I} X_{ijk} = 1, \ \forall j, k \tag{2.40}$$

$$X_{ijk} \leq \sum_{l \in L} W_{ikl}, \ \forall i, j, k \tag{2.41}$$

$$\sum_{l \in L} W_{ikl} \leq Y_i, \ \forall i, k \tag{2.42}$$

$$\lambda_{ik} = t_{ik} \sum_{j \in J} d_{jk} X_{ijk}, \ \forall i, k \tag{2.43}$$

$$R_{ik0} = 0, \ \forall i, k \tag{2.44}$$

$$a_{kn} - \lambda_{ik} \geq M_1 (R_{ikn} - 1), \ \forall i, k, n \tag{2.45}$$

$$a_{kn} - \lambda_{ik} \leq M_2 R_{ikn}, \ \forall i, k, n \tag{2.46}$$

$$Q_{ikn} = R_{i,k,n} - R_{i,k,n-1}, \ \forall i, k, n \tag{2.47}$$

$$V_{ikln} \leq M_3 Q_{ikn}, \ \forall i, k, l, n \tag{2.48}$$

$$V_{ikln} \leq M_3 W_{ikl},\ \forall i,k,l,n \quad (2.49)$$

$$V_{ikln} \geq 0\ \forall i,k,l,n \quad (2.50)$$

$$V_{ikln} \leq \sum_{j \in J} \frac{\delta_{ij} d_{jk} X_{ijk}}{d_k} b_{kln},\ \forall i,k,l,n \quad (2.51)$$

$$V_{ikln} \geq \sum_{j \in J} \frac{\delta_{ij} d_{jk} X_{ijk}}{d_k} b_{kln} - M_3(1 - Q_{ikn}) - M_3(1 - W_{ikl}),\ \forall i,k,l,n \quad (2.52)$$

$$\sum_{i \in I} \sum_{l \in L} \sum_{n \in N} V_{ikln} \geq \alpha_k,\ \forall k \quad (2.53)$$

$$Y_i, W_{ikl}, Q_{ikn}, R_{ikn} = 0 \text{ or } 1,\ \forall i,k,l,n. \quad (2.54)$$

The formulation has about $I(1+K(L+2N+LN+J)$ variables (which is on the order of $O(IK(J+LN)))$, and $K + JK + IK(J+3+3N+5LN)$ constraints (which is on the order of $O(IK(J+LN))$. For example, with 10 facilities, 20 customers, 2 parts, 5 stock levels, and 5 fill rate function steps, the formulation has 1210 variables and 3302 constraints. For a problem instance with 25 facilities, 150 customers, 10 parts, 5 stock levels, and 10 fill rate function steps, this formulation has 55025 variables and 106010 constraints.

This rather large, but linear, integer programming model captures all the cost trade-offs and the complex relationships among different entities of the overall problem. Taking advantage of the low-demand nature of the parts, the linearization process makes this possible, but it comes with a cost: The fill rate approximation used in the process is indeed an approximation, hence when the demands are allocated and stock levels are computed, the actual fill rates may be different from the fill rates approximated in the model. In fact,

there are two possibilities: (1) There is potential for an overestimation of fill rate in the model, in which the solution may not actually satisfy the service level constraints (hence, may become infeasible) when the actual fill rates are considered. (2) There is also potential for underestimation by the model, in which the solution may require overstocking, increasing costs unnecessarily. To resolve these issues, we add a post processing stage, as explained next.

2.2.4 Post Processing

We give two examples to illustrate the challenges of fill rate approximation and linearization, one for service constraint violation and another for overstocking. In one problem instance, the linearized model gives a solution in which the required level of 70% service is barely satisfied by opening two facilities, and stocking two units, one at each facility. The actual service level prescribed by model (2.39) - (2.54) is 1% short of the required level of 70%. Keeping the location and allocation decisions the same, we compute that we need one more part at one of the open facilities to achieve the target service level. In another instance, the linearized model calculates that a 68% service level (for required 50%) is achieved by holding a total of two parts. With the prescribed location, allocation, and stock level decisions, the actual service level is 69%, a 19% overshoot of the required 50%. We can actually keep the same open facilities and demand allocations, but shift inventory between locations, and still provide the required service level, eventually reducing the total cost by 400.

We report more on this issue later when we discuss the computational results in Section 2.3. Here we formalize the process of uncovering the potentials for revising stock levels with a post processing stage. This post processing stage will revise the stock levels, if necessary, (1) to guarantee ultimate feasibility of the solution (for service constraints) with minimal cost increase, and (2) to save in inventory costs as much as possible without sacrificing actual service (that is, still satisfying the required service level). To do this, we take the location and allocation decisions of the solution to the linearized model (2.39) - (2.54), and solve a multi-location inventory model to revise the stock level decisions (only if necessary) at the open facilities. As expected, this model borrows constraints from the linearized model, and is significantly smaller and simpler.

Suppose we solve the linearized model (2.39) - (2.54), and obtain the optimal solution with location decisions $Y_i = \hat{Y}_i$ for all i, demand allocation decisions $X_{ijk} = \hat{X}_{ijk}$ for all i, j, and k. As we focus on the open facilities only at this stage, we define $\hat{I}$ to be the set of open facilities. Once these decisions are known (and fixed), we can compute the (actual) mean lead time demand for each part at the open facilities:

$$\hat{\lambda}_{ik} = t_{ik} \sum_{j \in J} d_{jk} \hat{X}_{ijk}, \ \forall i \in \hat{I} \tag{2.55}$$

Moreover, we can now compute the actual fill rates:

$$\hat{\beta}_{ik}(l) = G_{ik}(l-1) = \sum_{r=0}^{l-1} \frac{(\hat{\lambda}_{ik})^r e^{-\hat{\lambda}_{ik}}}{r!} \tag{2.56}$$

where $G_{ik}(.)$ is the Poisson cumulative distribution function with mean lead time demand $\hat{\lambda}_{ik}$ for part k, calculated at each potential stock level l, $l = 0, 1, 2, \ldots, L$, at each open facility $i \in \hat{I}$. We then use these actual fill rates to rewrite the service level constraints, and decide if we need to revise the stock level decisions.

The following is a listing of the multi-location inventory model, whose only set of decision variables is the stock levels, which are modeled using the previously defined W_{ikl} variables.

$$\min \sum_{i \in \hat{I}} \sum_{k \in K} \sum_{l \in L} l h_{ik} W_{ikl} \tag{2.57}$$

$$\sum_{l \in L} W_{ikl} \geq \hat{X}_{ijk}, \ \forall i \in \hat{I}, j \in J, k \in K \tag{2.58}$$

$$\sum_{i \in \hat{I}} \sum_{j \in J} \sum_{l \in L} \frac{\delta_{ij} d_{jk} \hat{X}_{ijk}}{d_k} \hat{\beta}_{ik}(l) W_{ikl} \geq \alpha_k \ \forall k \in K \tag{2.59}$$

$$W_{ikl} = 0 \text{ or } 1, \ \forall i \in \hat{I}, k \in K, l \in L. \tag{2.60}$$

The objective (2.57) of the multi-location inventory model is to minimize the total inventory costs; the network design decisions are held fixed as prescribed by (2.39) - (2.54). Constraints (2.58) are forcing constraints to keep a part's stock at the open facilities serving that part. The service level constraints (2.59) are much simpler, because the network design decisions are held fixed.

The integrated solution approach is then the combination of the network design (location and demand allocation) decisions from the linearized

formulation and the inventory stocking decisions from the multi-location inventory formulation (if different from the linearized model's solution). The total cost can be calculated accordingly. We now report our computational results and insights.

2.3 Computational Study - Integrated Approach

2.3.1 Experimental Data Set

We conduct extensive tests to obtain insights about the behavior of the integrated model, the effect of post processing, and overall performance of the model. Later, we introduce another model that is analogous to the conventional approach of making the network design decisions first and then finding the stock levels (called *the decoupled approach*) to show the benefits of the integrated model.

We take our industrial partner's real data as a basis for our problem instances. The original data set includes different size problems of up to 20 candidate facilities, 150 customers, and 10 parts. Here, we report the results of problem instances for the following data set:

- **Candidate facilities (I) and customers (J)**: We use problem instances with 16 candidate facilities and 134 customers which are given by their zip codes in the real data.
- **Parts (K) and part demands (d_{jk})**: There are four parts with different demand patterns. Mean demands of each customer for each part

are given in the real data. To isolate the effects of different factors, we experiment with the single-part instances first by analyzing each demand pattern separately. Later, we also show results from experiments with all 4 parts' demands considered at the same time.

- **Fixed facility costs (f_i), inventory holding costs (h_{ik}), and transportation costs (c_{ijk})**: We generate these costs randomly from discrete uniform distributions and scale them with cost factors to control their relative weights in the objective function. Fixed costs are generated as follows:

$$f_i \sim \theta_F U[500, 1500]$$

where θ_F is the *fixed facility cost multiplier* that will be modified through the experiments. We use three different values for θ_F: 1, 10, and 100. Note that average fixed cost is 1000 when $\theta_F = 1$, and 100,000 when $\theta_F = 100$.

Similarly, we calculate the holding costs using

$$h_{ik} \sim \theta_H U[0.25 \times v_k/2, 0.25 \times 3v_k/2]$$

where θ_H is the inventory holding cost multiplier to be modified to create different experimental settings, and v_k is the unit cost of part k (modified from the real data). We use two different values for θ_H: 1 and 10.

Transportation costs are given in the real data, which are loosely based on distances between facilities and customers, and sometimes on part

specifications. We scale these values with the transportation cost multiplier θ_T which is either 1 or 10.

The cases with low fixed facility costs and/or high holding costs may seem unrealistic, but we should note these are instances with up to four parts where inventory holding costs can be magnified only through their unit holding costs. Most real SPL systems service thousands of high-cost parts, added together becoming a significant part of the overall cost. We tried to achieve a level of realism with similar compositions of the total costs by changing the cost multipliers. Overall, we generate three random instances using these distributions for each setting of cost multipliers θ_F, θ_T, and θ_H.

- **Time windows (w)**: We use two different settings for time windows: 2 hours and 4 hours, which are typical in real SPL systems. The time window identifier δ_{ij} for each facility and customer pair is given in the real data.
- **Replenishment lead times (t_{ik})**: We assume a lead time of one week for all facilities and parts, which, again, given by the real data.
- **Maximum stock level (L)**: We note that the maximum possible lead time demand (calculated from real data assuming all customer demands for a part are assigned to one virtual location) is quite variable, changing between 0.2 and 0.8 for different parts. This allows us to use a maximum stock level (L) of 5, at which the fill rate for even a virtual facility serving all possible demand is practically 100%.

Table 2.2: Experimental design

Factor	Description	Number of levels	Levels
θ_F	facility fixed cost multiplier	3	1, 10, 100
θ_H	inventory holding cost multiplier	2	1, 10
θ_T	transportation cost multiplier	2	1, 10
w	time windows	2	2 and 4 hours
α	target service levels	4	0.1, 0.3, 0.5, 0.7
K	demand patterns	4	
	number of instances	3	
	total number of instances:	1152	

- **The fill rate approximation (β_{ik})**: To facilitate the linearization of fill rates for a part, we divide the demand axis (0 to maximum possible lead time demand for the part) into $N = 10$ intervals. Since the most relevant demand values for actual facilities will be in the region with very low lead time demand, we use more and smaller intervals in that area, hence making wider intervals as demand increases. More specifically, we divide the demand axis as $a_{kn} = d_k/v^{(N-n)}$ for $n = 0, 1, \ldots, N$ to determine the break points of each step in fill rate approximation (See Figure 2.1 for an example). According to our preliminary experiments, we found out that $v = 1.5$ provides a good approximation.

- **Time-based service levels (α_k)**: We use the same target service level for all parts. We vary the target service levels for all parts (10%, 30%, 50% and 70%) in the experiments.

2.3.2 Effects of Post Processing

We first present results on the effects of post processing. We seek to know if there is significant need for this stage, and if so, how much revision/correction it does. Recall that the main reason for the post processing stage is the fill rate approximation utilized to linearize the original integrated model. This mainly affects the service level constraints. After solving the linearized model using the approximate fill rates (based on the demand intervals and step functions used for each stock level), we compute the actual fill rates (based on the actual demand assigned to each facility). Using these actual fill rates ($\hat{\beta}$), we then compute the (actual) time-based service levels ($\gamma_k(\hat{\beta})$), and compare them to the required service levels (α_k). Due to the mid-point approximation of the fill rates, it is possible for the model to under-estimate and over-estimate (actual) fill rates. Hence, we count the instances for which the model overestimates the actual fill rates and potentially causes infeasibility (for which the post processing stage has to reallocate or increase the stock levels), and the instances where the model underestimates the actual fill rates and potentially overstocks for which the post processing stage reallocates or decreases the stock levels to save inventory holding costs.

More specifically, we compute the actual service level for part k as follows:

$$\gamma_k(\hat{\beta}) = \sum_{i \in I} \sum_{j \in J} \frac{\delta_{ij} d_{jk} \hat{X}_{ijk}}{d_k} \hat{\beta}_{ik} \tag{2.61}$$

where $\hat{\beta}_{ik}$ is the actual fill rate of the linearized model's solution for part k at

facility i, calculated using the optimal demand allocation $\hat{X}_{ijk}$ for all j, k and the optimal stock level at facility i (for which $\hat{W}_{ikl} = 1$ for optimal stock level l for part k at location i). Hence, $\gamma_k(\hat{\beta})$ is the left hand side of constraints (2.18) for part k calculated using the actual fill rates, demand allocations and stock levels, obtained from the linearized model solution. We similarly calculate the model service level $\gamma_k(\beta)$ using the model's approximated fill rates (β_{ik}'s) for all facilities.

From Table 2.3, we categorize the results into two groups:

1. Comparing the model service levels ($\gamma_k(\beta)$) and the actual service levels ($\gamma_k(\hat{\beta})$) (as shown on the left hand side of Table 2.3), we observe the following:

- In almost half of the instances (542 out of 1152 or 47%), the linearized model overestimates the service level, $\gamma_k(\beta) > \gamma_k(\hat{\beta})$. In 610 out of 1152 instances (or 53%), the linearized model underestimates the service level, $\gamma_k(\beta) < \gamma_k(\hat{\beta})$.

- In both over- and under-estimated instances, the average absolute deviation between the model's approximated service level and the actual service level is less than 1%, showing the precision in the approximation of fill rates.

2. Comparing the actual service levels ($\gamma_k(\hat{\beta})$) and the target service levels (α_k) (as shown on the right hand side of Table 2.3), we observe the following:

- In almost 80% of the instances (919 out of 1152), the linearized model's solution satisfies the time-based service level, $\gamma_k(\hat{\beta}) > \alpha_k$. Each instance here belongs to one of two subcategories: $\gamma_k(\beta) > \gamma_k(\hat{\beta}) > \alpha_k$, or $\gamma_k(\hat{\beta}) > \gamma_k(\beta) \geq \alpha_k$. In only one of these instances (falling in the second category), the linearized model's solution is revised via post processing by reallocation of stock levels.

- In 233 out of 1152 instances (20.4%), the linearized model's solution cannot achieve the required service level so that $\gamma_k(\hat{\beta}) < \alpha_k$. This is the category of solutions for which post processing pays off well. To see the amount of violation of service level constraints, we also list the "average absolute deviation from the target service level" in the table (% Dev'n). We calculate this amount by taking the average of the absolute differences between the actual service level $\gamma_k(\hat{\beta})$ and the target service level α_k. For this category, the average deviation is around 0.7%. The distribution of these 233 instances with respect to the amount of violations is depicted in Figure 2.2. We note that the amount of violation is less than 1% in 174 of these instances, which shows that the fill rate approximation causes some service level violations, but the amount of violation is extremely small in the majority of instances.

Table 2.3: Comparison between the model service levels and actual service levels, and the effects of post processing

$\gamma_k(\beta)$ vs. $\gamma_k(\hat{\beta})$					$\gamma_k(\hat{\beta})$ vs. α_k			
Over/under-estimate	Ins.	% Dev'n	Instances		Feasibility	Ins.	% Dev'n	Rev'd Ins.
Over: $\gamma_k(\beta) > \gamma_k(\hat{\beta})$	542^*	0.52	$\gamma_k(\beta) \geq \alpha_k > \gamma_k(\hat{\beta})$	233	Infeasible: $\gamma_k(\hat{\beta}) < \alpha_k$	233	0.68	233
			$\gamma_k(\beta) > \gamma_k(\hat{\beta}) > \alpha_k$	309^*	Feasible: $\gamma_k(\hat{\beta}) > \alpha_k$	919	7.01	1
Under: $\gamma_k(\beta) < \gamma_k(\hat{\beta})$	610	0.83	$\gamma_k(\hat{\beta}) > \gamma_k(\beta) \geq \alpha_k$	610				

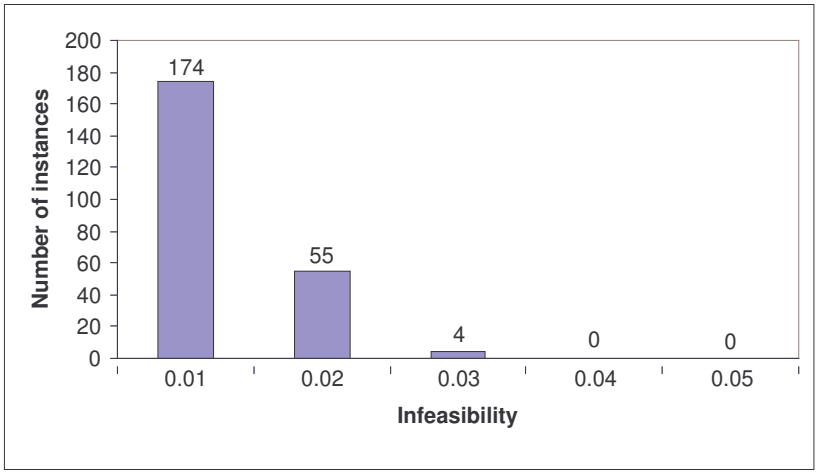
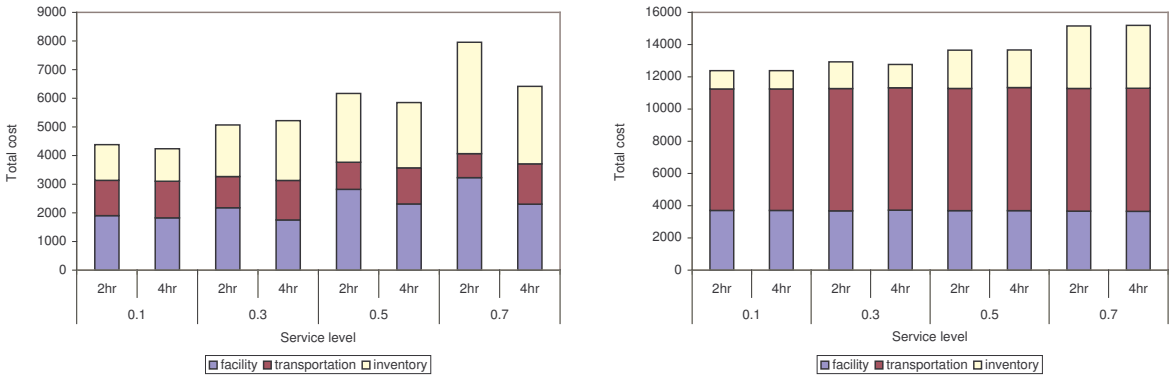


Figure 2.2: Histogram of amount of service level violations (infeasibilities) for the linearized model

2.3.3 Integrated Approach Costs

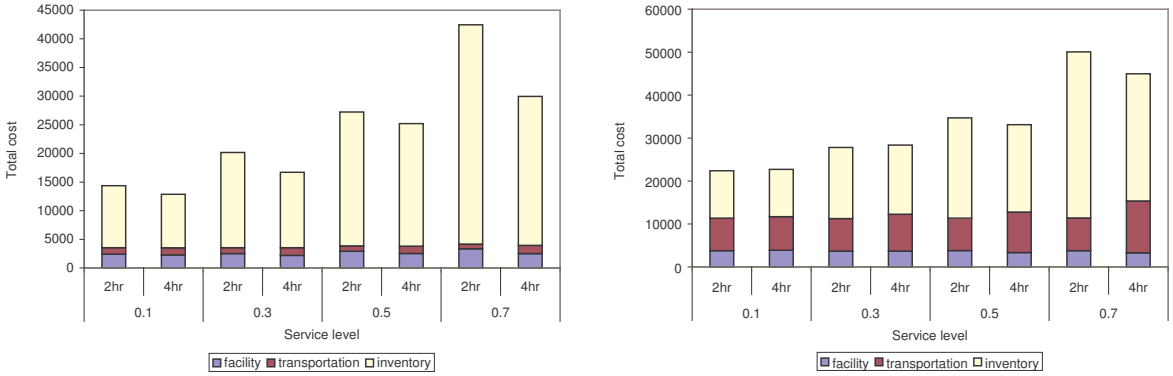
We also investigate the changes in the costs as we change the cost multipliers, service windows, and service level requirements. All costs are calculated after post processing.

In Figures 2.3-2.5, each column plots the average cost of 12 instances (4 demand patterns with 3 randomized facility-holding cost realizations). As expected, total costs increase with increasing target service levels (α_k). This effect is magnified for instances with 2-hour service windows as compared to 4-hour window instances. As expected, a longer time window is less restrictive, providing opportunities to achieve lower cost solutions. Also, as expected, the total cost increases significantly as we increase the cost multipliers. Along with them, the composition of the total cost among the three components (fixed



(a) $\theta_T = 1$, $\theta_H = 1$

(b) $\theta_T = 10$, $\theta_H = 1$

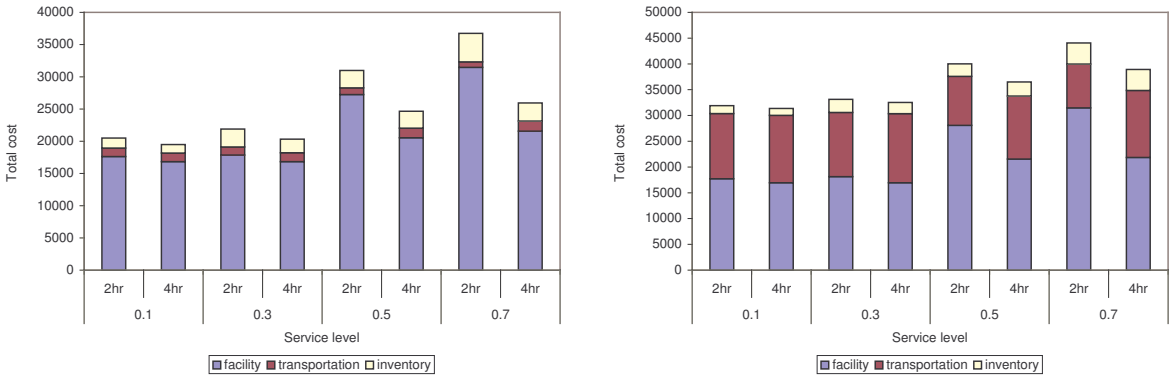


(c) $\theta_T = 1$, $\theta_H = 10$

(d) $\theta_T = 10$, $\theta_H = 10$

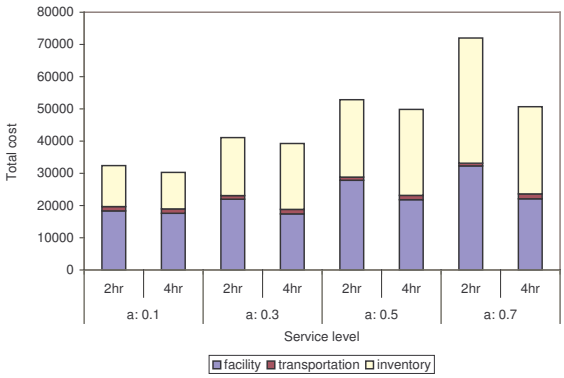
Figure 2.3: Integrated model costs for instances with low fixed facility costs ($\theta_F = 1$)

facility, transportation, and inventory) differs quite visibly with the changes in the cost multipliers. In general, as a particular cost multiplier increases, its component's share of total cost goes up, and after certain level its component dominates the overall costs. This is especially true for the fixed facility costs, as when its multiplier is set at 100, the shares of inventory and transportation

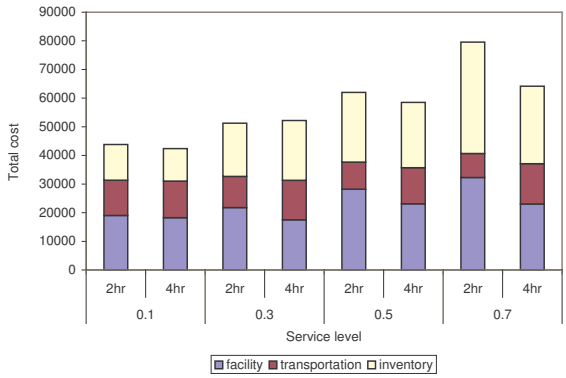


(a) $\theta_T = 1$, $\theta_H = 1$

(b) $\theta_T = 10$, $\theta_H = 1$



(c) $\theta_T = 1$, $\theta_H = 10$

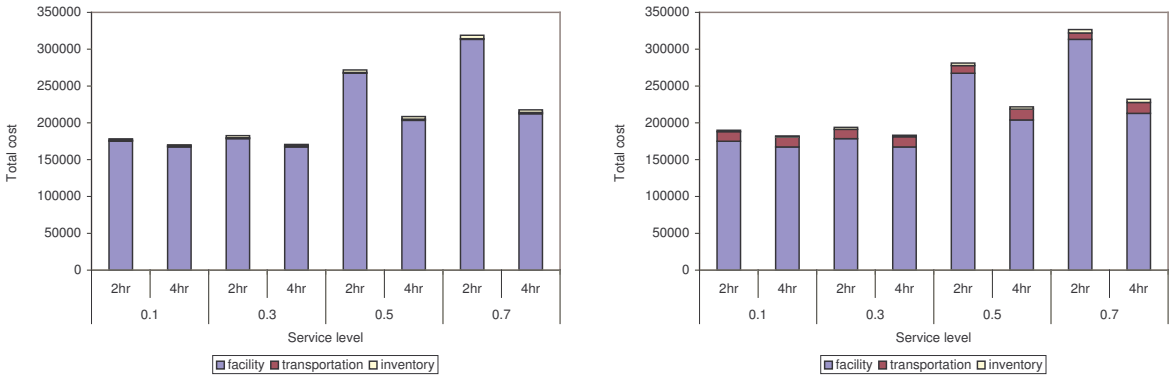


(d) $\theta_T = 10$, $\theta_H = 10$

Figure 2.4: Integrated model costs for instances with medium fixed facility costs ($\theta_F = 10$)

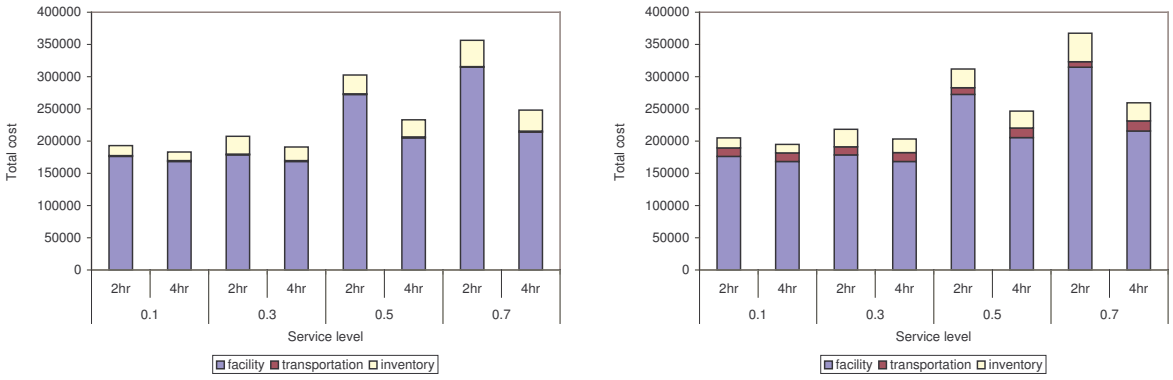
components become marginal.

The total cost also goes up as we increase the target service levels. This is mainly due to an increase in facility (fixed) costs as the model is forced to open more facilities to satisfy increasing time-based service levels, and partially due to higher stock levels needed to support higher service levels.



(a) $\theta_T = 1$, $\theta_H = 1$

(b) $\theta_T = 10$, $\theta_H = 1$



(c) $\theta_T = 1$, $\theta_H = 10$

(d) $\theta_T = 10$, $\theta_H = 10$

Figure 2.5: Integrated model costs for instances with high fixed facility costs ($\theta_F = 100$)

(The solutions suggest opening about 3 facilities (out of 16) on average over all problem instances, requiring a minimum of 2 facilities for low service levels and 4-hr time windows, and up to 5 for higher service levels and 2-hr time windows). Transportation costs exhibit contrasting behavior since, as we open more facilities due to increased service level, the distances from facilities to

customers decrease. This translates into reductions in transportation costs.

One would expect the total costs for instances with the 2-hour service window to be always greater than the total costs for instances with the 4-hour service window, for the same cost multipliers and the target service level. However, we observe three exceptions to this in our experiments. As shown in Figure 2.3.a at $\alpha = 0.3$, Figure 2.3.d at both $\alpha = 0.1$ and $\alpha = 0.3$, and Figure 2.4.d at $\alpha = 0.3$, the opposite is true, i.e., the total cost of a 4-hr service window is higher than that of a 2-hr window. This is mainly due to the revision performed at the post processing stage. We note that the post processing revision takes place more often and more significantly with the 4-hr service window. Especially for the low target service levels, with 4-hr windows, average demand allocation per facility increases, hence the cost of overestimating fill rates is high, which leads the post processing stage to increase the stock levels in more instances. We note that the holding costs in these unexpected settings are higher than those of 2-hr service window problems.

2.3.4 Lower Bound and Quality of Fill Rate Approximation

Because of the nonlinear nature of the original formulation, it is not possible to solve even small problem instances to optimality. Even if solved by available state-of-the-art nonlinear solvers, we are not guaranteed to obtain the global optimal solutions. Hence, it seems that the only reasonable ways to measure the quality of our fill rate approximation and its associated

solution are (1) to find a lower bound solution and measure the gap between the lower bound and the reported solution of the approximated model, and (2) to increase the accuracy of the fill rate approximation and show how much improvement could be obtained with the increased accuracy. To do (1), we use an overestimating fill rate approximation in the linearized model, i.e. define the step-function based approximation with (optimistic) maximum points at each step (not with mid-points of the intervals of the original fill rate function as in Figure 2.1). In a way, we draw the approximating step function above the actual fill rate curve. To do (2), we solve the linearized model with two more levels of the number of steps in the approximation ($N = 5$ and $N = 15$), along with $N = 10$ steps.

Table 2.4 reports the linearized model costs for different levels of fill rate approximations (with 5 steps, 10 steps and 15 steps), and the percentage gaps between the tightest lower bound solution and the integrated model solution (obtained with 10-step fill rate approximation). The experiments are conducted for one demand pattern (A), one instance (data set 1), with 4-hour service time window, and 70% service level requirement. We limit the solution times to two hours, and the instances shown with * are not finished within this time limit (hence not guaranteed to be optimal).

Table 2.4: Percentage gap between lower bound and integrated model solutions

	$\theta_H = 1$						$\theta_H = 10$						
	$\theta_T = 1$			$\theta_T = 10$			$\theta_T = 1$			$\theta_T = 10$			
	θ_F			θ_F			θ_F			θ_F			
	1	10	100	1	10	100	1	10	100	1	10	100	Time
$N = 5$	3970	16251	133521	11841	26266	143556	20349	32076	149346	26513	39797	162508	7.3
$N = 10$	3841	15604	132874	11415	26263	143590	13853	25580	142850	25166	38406	156044	369.9
$N = 15$	3838	15602	132872	11816*	26244	143549	13850	25577	142847	30500*	38379	156017	1426.7
LB	3835	15599	132869	11305	25781	143517	13441	25168	142438	25048	38352	155990	
% Gap	0.14	0.03	0.00	0.97	1.87	0.05	3.06	1.64	0.29	0.47	0.14	0.03	

As expected, as we increase the accuracy of the fill rate approximation, we obtain better solutions from the linearized model (hence find solutions closer to the lower bound, except the two cases where the 15-step model is not solved to optimality). The 10-step approximation seems to provide the best compromise between the solution quality and computational time by providing significantly improved solutions as compared to the 5-step approximation, still staying reasonable in terms of the computational time. The improvement with the 15-step approximation does not seem to be significant, and the limited improvement comes with an unbearable increase in computation time. Hence, we use the 10-step approximation in our experiments.

Moreover, the reported solutions of the 10-step approximation are mostly within 1% of the lower bound, signaling the fact that the solutions we provide are near-optimal. The additional improvement that could be obtained by a more accurate approximation is bounded by these percentage gaps. Hence, we think that this is sufficient as a first step to show the efficacy of the proposed linearized approach with the step-function-based approximation.

2.4 Decoupled Approach

Another goal of this study is to show the benefit of a model that integrates network design and inventory stocking. We now introduce the "decoupled model" that we use to emulate the conventional and practical approach of sequentially deciding the network design first and choosing the stock levels later.

In fact, the decoupled model actually consists of two submodels: (1) Logistics network design submodel (LND-only submodel), and (2) Inventory stocking submodel (IS-only submodel). The LND-only submodel locates the facilities of the network and allocates demands to these facilities without considering inventory costs and assuming practically free 100% fill rate for all parts at all (open) facilities. The IS-only submodel takes the network and the demand allocations as input and decides the stock levels for all facilities and parts while trying to satisfy time-based service levels.

Both submodels borrow notation from the integrated model. In fact, the separation of the two in the decoupled model simplifies the service level constraints, and makes both submodels integer linear programming problems. Hence, we introduce the submodels without defining new notation or decision variables.

The LND-only submodel is a version of the service-constrained, multi-commodity uncapacitated facility location problem, which is practically the same as the one used to introduce the integrated model earlier in Section 2.2.2:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} d_{jk} X_{ijk} \tag{2.62}$$

$$\sum_{i \in I} X_{ijk} = 1, \quad \forall j \in J, k \in K \tag{2.63}$$

$$X_{ijk} \leq Y_i, \quad \forall i \in I, j \in J, k \in K \tag{2.64}$$

$$\sum_{i\in I}\sum_{j\in J}\frac{\delta_{ij}d_{jk}X_{ijk}}{d_k} \geq \alpha_k,\ \forall k \in K \tag{2.65}$$

$$0 \leq X_{ijk} \leq 1,\ \forall i \in I, j \in J, k \in K \tag{2.66}$$

$$Y_i = 0 \text{ or } 1,\ \forall i \in I. \tag{2.67}$$

Since we have discussed the objective function and constraints previously, we provide no further description here. From the optimal solution of the LND-only submodel, we obtain $\tilde{I}$ as the set of open facilities ($\tilde{I} = \{i \in I : Y_i = 1\}$) and the demand allocation decisions $\tilde{X}_{ijk}$ for all i, j, and k. Using these, we calculate the actual mean lead time demands for all open facilities and parts:

$$\tilde{\lambda}_{ik} = t_{ik}\sum_{j} d_{jk}\tilde{X}_{ijk},\ \forall i \in \tilde{I}, k \in K. \tag{2.68}$$

We then compute the actual fill rates for all parts and facilities, for each potential stock level:

$$\tilde{\beta}_{ik}(l) = G_{ik}(l-1) = \sum_{l'=0}^{l-1}\frac{(\tilde{\lambda}_{ik})^{l'}e^{\tilde{\lambda}_{ik}}}{l'!}. \tag{2.69}$$

These all become input to the IS-only submodel, which is itself a multi-facility inventory model, where the only decision variables are the stock levels, W_{ikl}'s. The submodel finds the minimum-cost stock levels possible to achieve time-based service levels using the calculated actual fill rates. As the network and demand allocations are decided and fixed already, the remaining objective is to minimize the total inventory costs. In this respect, the IS-only submodel is similar to the post-processing model of the integrated approach:

$$\min \sum_{i\in\tilde{I}}\sum_{k\in K}\sum_{l\in L} lh_{ik}W_{ikl} \tag{2.70}$$

$$\sum_{l \in L} W_{ikl} \geq \tilde{X}_{ijk}, \ \forall i \in \tilde{I}, j \in J, k \in K \tag{2.71}$$

$$\sum_{i \in \tilde{I}} \sum_{j \in J} \sum_{l \in L} \frac{\delta_{ij} d_{jk} \tilde{X}_{ijk}}{d_k} \tilde{\beta}_{ik}(l) W_{ikl} \geq \alpha_k \ \forall k \in K \tag{2.72}$$

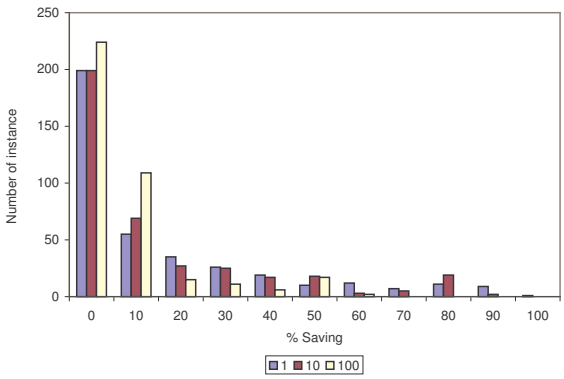
$$W_{ikl} = 0 \text{ or } 1, \ \forall i \in \tilde{I}, k \in K, l \in L. \tag{2.73}$$

Combining the network design solution from the LND-only submodel and the inventory stocking decisions from the IS-only submodel, we compute the total cost of the decoupled approach as the sum of the two objectives in the submodels.

2.5 Integrated versus Decoupled

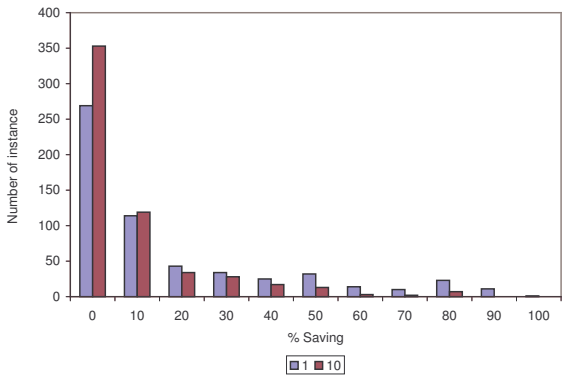
We now compare the two approaches to quantify their benefits and drawbacks (if any). First, we show how much one saves in total costs by using the integrated approach instead of the decoupled approach and investigate how these savings change with different problem settings and conditions. We then compare their actual solutions, namely, resulting networks (number and locations of open facilities, demand allocations and inventory decisions) to better understand the sources of the savings.

Figure 2.5 shows the histograms of the saving percentages of the integrated approach over the decoupled approach for each specific set of cost coefficients. The saving percentage is calculated as 100 times the total cost difference between the integrated and the decoupled approaches divided by the decoupled approach's total cost.



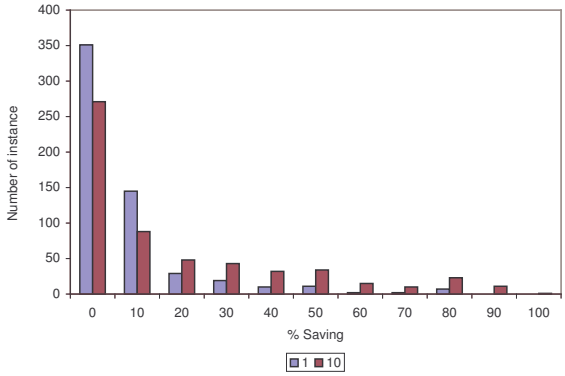
θ_F

(a) Effects of varying θ_F



θ_T

(b) Effects of varying θ_T



θ_H

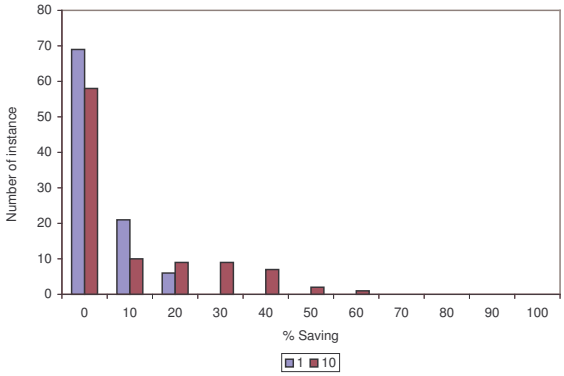
(c) Effects of varying θ_H

Figure 2.6: Cost savings obtained via the integrated approach over the decoupled approach for varying levels of the cost multipliers

The integrated approach achieves arguably significant cost savings when the fixed facility and transportation costs are relatively low and/or the holding costs are relatively high. As in Figure 2.7.c, there are some instances (almost 10 of them) achieving cost savings of more than 80% for $\theta_H = 10$. Inter-

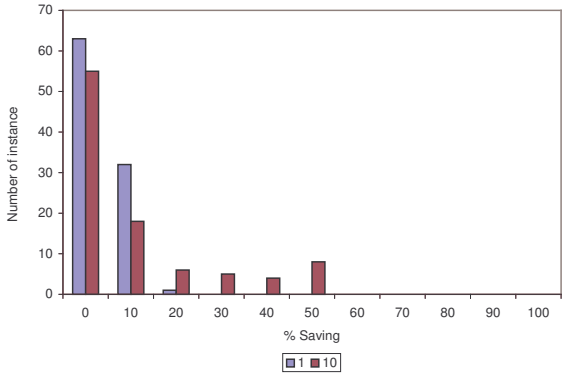
preted in another way, the decoupled approach emulating the current practice of sequentially designing the network and then prescribing the stock levels may be up to 5 times the cost of making the decisions concurrently. The savings are lower when the opposite conditions are present. When the facility and transportation costs are high and holding costs are low, the overall problem resembles the uncapacitated service-constrained facility location problem, which means inventory stocking loses its importance. In these cases, the decoupled approach produces quite accurate decisions in the LND-only submodel while ignoring inventory. We can see this behavior of the models in Figure 2.7, in which we fix the facility and transportation costs to various combinations of the multipliers' levels. Each graph shows the histogram for both low and high holding costs. From these, we can easily state that the decoupled approach fails more often and more significantly when inventory costs are larger.

As shown in Figure 2.8, the integrated approach yields more savings for instances with a 4-hr time window than those with a 2-hr time window. When time windows are longer, the decoupled approach tends to assign more demand to each facility, potentially using fewer facilities (depending on the fixed facility costs), without considering inventory implications of these decisions. Using the LND-only solution, the IS-only submodel tries to achieve very high fill rates for each of these facilities with high demands to be able to satisfy the service levels. With more demand assigned per facility, the IS-only submodel can do this only by keeping high stock levels, causing total cost to increase significantly. The integrated approach in these settings is able consider all these interactions by



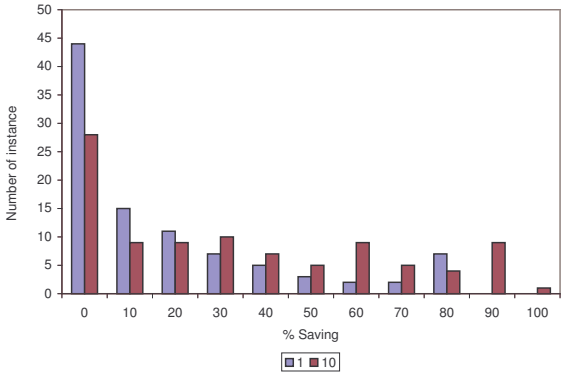
θ_H

(a) Fixed levels of $\theta_F = 1$, $\theta_T = 10$



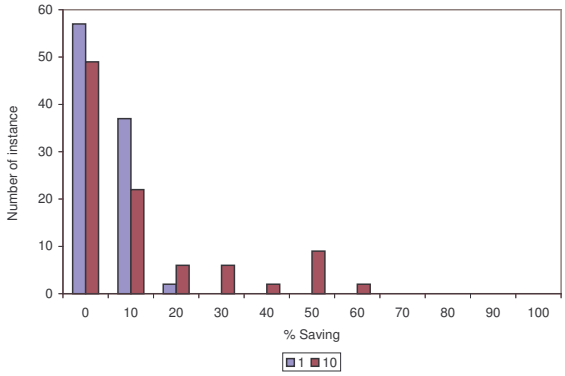
θ_H

(b) Fixed levels of $\theta_F = 100$, $\theta_T = 10$



θ_H

(c) Fixed levels of $\theta_F = 1$, $\theta_T = 1$



θ_H

(d) Fixed levels of $\theta_F = 100$, $\theta_T = 1$

Figure 2.7: Cost savings obtained via the integrated approach over the decoupled approach for varying levels of the cost multipliers (two levels of fixed facility and two levels of transportation costs), where θ_H is varied in each plot.

design and finds the best trade-off between all cost components and service constraints.

As summary of the single-part results, Figure 2.9 shows the trade-off

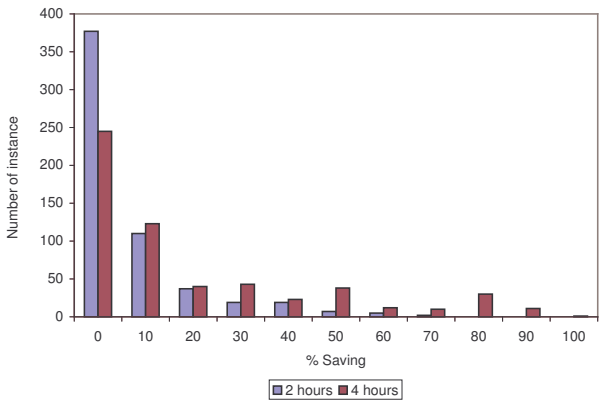
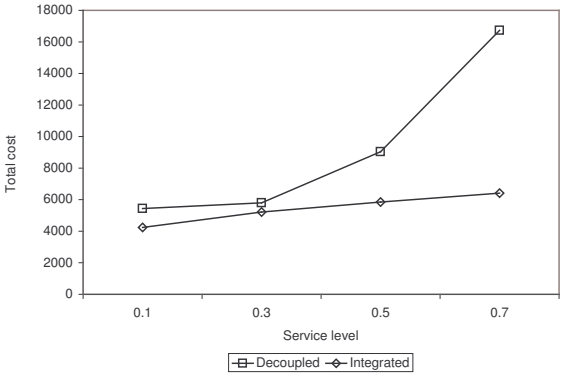


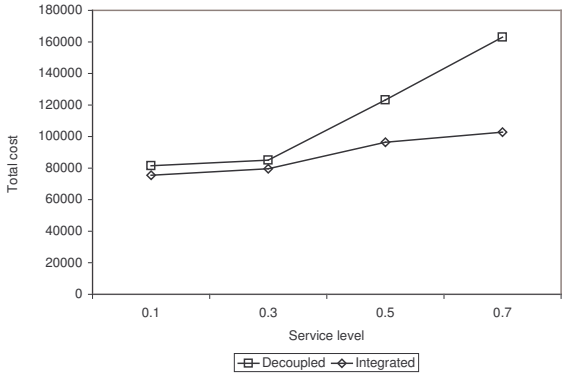
Figure 2.8: Cost savings obtained via the integrated approach for the tested time windows

curves (also called the "efficient frontiers") that draw (optimal) total cost versus service level for both approaches. Figure 2.9.a depicts the total costs averaged over instances with 4 hour time window and cost multipliers of $\theta_F = \theta_T = \theta_H = 1$. Figure 2.9.b averages the total costs of all instances with 4 hour time window (regardless of their cost multipliers). Figure 2.9.c averages the total costs of all 1152 instances. Note that there are tremendous opportunities for cost savings through integrating network design and inventory stocking.

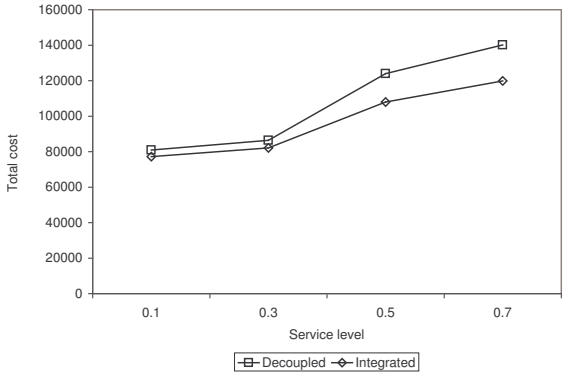
The figure shows that one can shift the whole trade-off curve to the right (superior regions) by integrating network design and stocking. For example, in order to achieve a 50% service level, the decoupled approach suggests that we need almost \$124,000 on average (see Figure 2.9.c). However, the integrated approach requires only \$108,000 to guarantee the same level of service. Similarly, with \$124,000 we can achieve 50% service level using the decoupled



(a) Time window w = 4-hr, $\theta_F = \theta_T = \theta_H = 1$



(b) Average of all instances with 4-hr windows



(c) Average of all 1152 instances

Figure 2.9: Trade-off curves for the single-part experiments (Each point is an average of total (optimal) costs all instances with given settings under each chart)

approach, while we can achieve more than 70% service level with the integrated approach with the same budget. As shown in the figure, the savings become larger for higher target service levels, achieving arguably significant cost savings at certain settings such as that of Figure 2.9.a.

We now investigate the conditions under which the solutions of the two approaches differ, eventually producing sometimes significantly different total costs. Tables 2.5 and 2.6 show the network differences between the integrated and the decoupled optimal solutions. In this table,

- 0 means the two produce totally different solutions (different numbers of open facilities, different demand allocations, and different stock levels),
- 1 means the number of open facilities is the same in both solutions, but everything else is different,
- 2 means both the number and the locations of open facilities are the same, but the demand allocations and the stock levels at these facilities are different between the two solutions,
- 3 means all aspects of the solutions are the same except the stock levels, and finally
- 4 means that the two solutions are exactly the same.

The table provides a lot of detail about the solutions of the two approaches. For example, the two approaches produce pretty much the same solution for the setting where $\theta_F = 100, \theta_T = 10$, and $\theta_H = 1$. However, the integrated approach finds a completely different solution (including the number and the locations of open facilities) for $\theta_F = 1, \theta_T = 1$, and $\theta_H = 10$. Under these conditions, using a decoupled approach, and, as a result, designing a suboptimal network may be costly, as the wrong network decisions will undermine

the system performance for a long time before a revision of the network is undertaken (and corrected).

We can ultimately calculate a measure of average solution similarity by taking an average of the values for each setting. For example, the average similarity within $\theta_F = 100, \theta_T = 10, \theta_H = 1$ is 2.81 for both demand patterns 1 and 3, whereas the average similarity within $\theta_F = 1, \theta_T = 1, \theta_H = 10$ is 1.59 for demand pattern 1 and even lower (0.81) for demand pattern 3. This shows the benefit of including multiple demand patterns, as they affect locations, allocations, and stock levels.

Our intuition is that the original model should resolve the multi-part case because real SPL systems support and stock multiple parts. Hence, we now present results of the instances in which we consider all 4 demand patterns at the same time, each applied to a different part. Instead of a detailed analysis of the multi-part cases, for brevity, we show the total costs vs. service level trade-off curves for multiple-part instances. Figure 2.10 draws trade-off curves for both the decoupled and integrated approaches for a 4-hr time window and by using data instance 1 (in total, we run 48 instances, each point on the graphs representing the average of 12 instances: 3 levels of fixed costs, 2 levels of transportation costs, and 2 levels of holding costs). The figure shows parallel results to the earlier ones and we make similar observations here as before.

Although our focus is on the overall quantification of benefits of considering inventory as part of the network design problem, we give an overall feel for its computational difficulty by listing computer times (in CPU seconds)

Table 2.5: Network comparison between the integrated and decoupled approaches

Data	Demand Patterns	w	α_k	$\theta_H=1$ $\theta_T=1$ θ_F 1	10	100	$\theta_T=10$ θ_F 1	10	100	$\theta_H=10$ $\theta_T=1$ θ_F 1	10	100	$\theta_T=10$ θ_F 1	10	100
1	A	2hr	0.1	2	2	2	2	2	4	2	2	2	2	2	2
			0.3	2	2	2	0	2	2	2	2	2	0	2	2
			0.5	1	4	4	4	1	2	1	1	2	4	1	4
			0.7	4	2	2	2	2	4	2	2	2	4	2	4
		4hr	0.1	2	4	4	2	4	4	1	4	4	1	4	4
			0.3	2	4	4	0	2	4	2	4	4	2	2	4
			0.5	1	2	2	2	1	2	1	2	2	0	1	2
			0.7	2	2	2	2	2	2	1	2	2	1	2	2
	B	2hr	0.1	4	2	2	4	4	2	4	4	4	4	2	4
			0.3	2	2	2	4	2	2	2	2	2	4	2	2
			0.5	4	4	4	4	4	4	4	4	4	4	4	4
			0.7	4	4	2	4	4	4	4	4	4	4	4	4
		4hr	0.1	2	2	2	4	2	2	1	2	2	4	2	2
			0.3	2	2	2	4	2	2	1	2	2	4	2	2
			0.5	4	1	2	4	4	2	4	1	2	4	4	1
			0.7	2	1	2	4	2	2	1	1	2	4	2	1
	C	2hr	0.1	4	1	2	2	2	4	0	1	1	1	2	1
			0.3	0	2	2	1	2	2	0	0	2	1	0	2
			0.5	1	1	2	1	1	2	1	1	1	1	1	1
			0.7	2	2	4	1	2	4	2	2	2	1	2	4
		4hr	0.1	2	1	4	2	4	4	0	1	1	1	4	1
			0.3	2	1	4	1	2	4	0	1	1	1	2	1
			0.5	0	4	4	1	2	4	0	0	4	0	0	4
			0.7	0	1	2	1	2	2	0	1	1	0	0	1
	D	2hr	0.1	1	4	2	4	4	2	1	1	2	1	1	4
			0.3	0	4	4	4	0	2	0	0	4	4	0	4
			0.5	4	4	4	4	4	4	4	4	4	4	4	4
			0.7	1	2	2	4	2	2	1	1	2	1	1	2
		4hr	0.1	2	2	2	4	2	2	2	2	2	0	2	2
			0.3	1	2	2	4	1	2	1	2	2	1	1	2
			0.5	4	4	2	4	4	4	4	4	1	4	4	4
			0.7	2	2	2	4	2	2	2	2	2	0	2	2
2	A	2hr	0.1	2	2	2	1	2	2	1	2	2	0	2	2
			0.3	2	2	2	4	2	2	2	2	2	4	2	2
			0.5	1	0	2	2	1	2	1	0	0	4	1	0
			0.7	4	2	2	4	2	2	2	0	0	4	4	4
		4hr	0.1	2	4	4	1	2	2	1	2	4	0	4	4
			0.3	2	2	2	4	2	2	1	2	2	0	2	2
			0.5	2	2	2	1	2	4	1	2	2	1	2	2
			0.7	2	2	2	1	2	2	2	2	2	0	2	2
	B	2hr	0.1	4	1	4	4	4	4	4	1	1	4	4	1
			0.3	1	4	4	2	4	4	1	1	4	2	1	4
			0.5	4	4	4	4	4	4	4	4	4	4	4	4
			0.7	1	2	2	4	2	2	1	1	2	4	1	2
		4hr	0.1	4	1	4	4	4	4	1	1	1	4	4	1
			0.3	4	1	2	4	4	1	1	1	1	4	4	1
			0.5	2	2	2	4	2	2	2	2	2	4	2	2
			0.7	2	1	2	4	2	1	1	1	1	4	2	1
	C	2hr	0.1	2	4	4	0	4	2	4	2	2	0	4	2
			0.3	2	2	2	4	2	2	0	2	2	4	2	2
			0.5	2	0	0	0	2	0	2	0	0	0	2	0
			0.7	2	2	4	4	4	2	2	2	2	4	2	4
		4hr	0.1	1	4	4	0	4	2	1	1	4	0	1	4
			0.3	2	2	2	0	2	2	2	2	2	0	2	2
			0.5	1	1	2	0	2	2	1	1	1	0	1	1
			0.7	1	1	2	0	2	2	1	1	1	0	1	1
	D	2hr	0.1	4	4	1	4	4	4	4	4	1	4	4	4
			0.3	4	4	1	4	4	4	4	4	1	4	4	4
			0.5	2	4	4	4	2	4	2	4	4	4	2	4
			0.7	1	2	2	4	2	2	1	1	2	4	1	2
		4hr	0.1	4	4	1	4	4	4	4	4	1	4	4	4
			0.3	2	2	1	4	2	2	2	2	1	4	2	2
			0.5	2	2	2	4	1	2	2	1	2	4	2	2
			0.7	1	2	1	4	1	2	1	2	1	0	1	2

Table 2.6: Network comparison between the integrated and decoupled approaches (Continued)

Data	Demand Patterns	w	α_k	$\theta_H = 1$						$\theta_H = 10$					
				$\theta_T = 1$			$\theta_T = 10$			$\theta_T = 1$			$\theta_T = 10$		
				θ_F			θ_F			θ_F			θ_F		
				1	10	100	1	10	100	1	10	100	1	10	100
3	A	2hr	0.1	2	2	4	2	4	2	0	2	4	2	4	2
			0.3	2	2	2	4	2	2	0	2	2	4	2	2
			0.5	2	4	2	2	4	2	1	2	4	0	2	2
			0.7	1	4	2	1	4	2	0	1	2	0	1	4
		4hr	0.1	1	4	4	2	4	2	1	1	2	0	1	2
			0.3	4	4	4	4	4	4	1	4	4	2	4	4
			0.5	2	2	2	2	2	2	1	2	2	2	2	2
			0.7	1	2	2	2	2	2	0	1	2	1	1	2
	B	2hr	0.1	4	4	4	4	4	4	0	4	4	2	4	4
			0.3	2	2	2	2	2	2	0	1	2	2	2	2
			0.5	4	4	4	2	4	4	4	4	4	2	4	4
			0.7	4	4	4	2	4	4	1	4	4	2	4	4
		4hr	0.1	4	4	4	2	4	4	1	4	4	1	4	4
			0.3	2	2	2	2	2	2	1	2	2	2	2	2
			0.5	4	2	2	2	4	2	4	1	2	2	4	2
			0.7	1	2	2	2	4	2	1	1	2	1	1	2
	C	2hr	0.1	0	2	2	2	4	4	0	1	2	2	0	2
			0.3	0	2	2	0	4	4	0	0	2	0	0	4
			0.5	1	1	4	0	1	2	1	1	1	0	1	1
			0.7	4	2	4	0	2	2	2	2	2	0	4	4
		4hr	0.1	0	2	2	2	2	4	0	1	2	2	0	2
			0.3	1	2	2	0	2	2	1	1	2	2	1	2
			0.5	0	2	2	0	2	2	0	2	2	0	0	2
			0.7	2	2	2	0	2	2	2	2	2	0	2	2
	D	2hr	0.1	1	4	4	2	4	4	0	1	4	2	1	4
			0.3	0	4	4	2	4	4	0	0	4	2	0	4
			0.5	1	4	4	2	4	4	1	1	4	2	1	4
			0.7	1	1	4	2	1	4	1	1	1	2	1	1
		4hr	0.1	1	4	4	2	4	4	0	1	4	2	1	4
			0.3	1	2	2	2	2	2	0	1	2	2	1	2
			0.5	1	1	2	2	4	2	1	1	1	2	1	1
			0.7	1	1	2	2	1	2	1	1	2	2	1	1

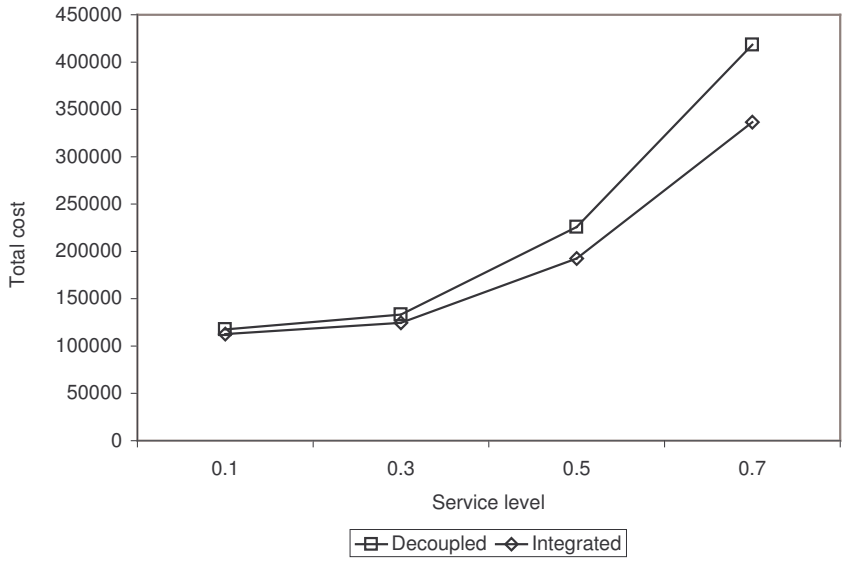


Figure 2.10: Trade-off curves for the multi-part instances (4-hr time window and data 1)

in Table 2.7. The table shows that, even for the relatively small problem instances considered in this chapter, the integrated model is a computationally challenging problem that needs further analysis, scalable solution techniques, and powerful heuristics. We will further focus on this idea in the fourth chapter.

2.5.1 Decoupled Approach with Varying Fill Rates

In the original decoupled approach, we had assumed that each facility has a 100% fill rate in the first step (location and allocation) and we solved for the required inventory levels in the second step. This was done to imitate the conventional approach of solving a coverage-based facility location allo-

Table 2.7: Computation times for the integrated model, post-processing, and decoupled model

	Integrated Model	Post Processing	Decoupled
Average	215.80	0.046	0.3
Min	0.48	0.03	0.094
Max	6461.95	0.33	2.69

cation model as the first step and then deciding the inventory levels for the given network. To see if improved solutions are possible within the decoupled approach, we introduce its refined version which solves the first step assuming that the open facilities will have lower (but still fixed) fill rates such as 95%.

Table 2.8 shows the average costs across 12 instances (demand patterns A, B, C, and D; instances 1, 2, and 3). All run with the cost coefficients $\theta_F = \theta_T = \theta_H = 1$; a 4-hour service window, and varying service level requirements (10% - 70%). We also run the decoupled approach, changing fill rate values used in the first step (as part of the coverage constraint in the LND-only model) from 100% to 80% in 5% increments. We compare these solutions with the integrated model solution, and show the gap between the best of the decoupled approach (obtained by picking the lowest cost among the costs of the different fill rates for each of the 12 instances) and the integrated approach.

As shown in the table, it is possible to improve the decoupled approach by trying different fill rates. However, it is still not possible to obtain solutions that are superior to those prescribed by the integrated model. In fact, we have an increasing improvement obtained from the integrated approach as compared

to the variable fill rate approach for the decoupled approach. Especially at high service levels (50% and 70%), we have significant improvements by integration, up to 16% compared to the best decoupled approach. We show the best performing fill rates in bold font for each service level requirement. Note that, different fill rates yield the best performance for different service levels, which may mean that one has to run many decoupled fill rates in the decoupled approach to find an overall good solution.

Table 2.8: Integrated model versus decoupled approach with different fill rates at first step

	Decoupled approach with fill rate (%)						
	100	95	90	85	80	Integrated	% gap
0.1	5437.2	**4334.24**	4334.82	4335.48	4336.21	4237.49	2.28
0.3	5798.18	5730.8	5583.06	5520.98	**5386.16**	5217.81	3.23
0.5	9039.47	6914.31	6917.7	**6350.45**	6354.88	5852.36	8.51
0.7	16731.3	8156.64	7939.91	**7450.63**	7867.07	6415.04	16.14

2.6 Conclusions

In this chapter, we consider an integrated approach to model both logistic network design and inventory stocking decisions with time-based service requirements, all in one monolithic formulation. The main challenge in this modeling effort is to capture the interactions between these sets of decisions so that the overall results are compatible and globally optimal. We conduct extensive computational tests using instances based on real-life data to quantify

the benefits of considering inventory explicitly as part of the network design model. Despite the modeling challenges and approximations of fill rate calculations introduced, the potential for cost savings from integration is apparent and the suboptimality of the decoupled approach in terms of the total logistics cost is undeniable.

We observe that we can achieve the same service levels at less cost with the integrated approach when compared with the traditional approach of making these decisions sequentially. Similarly, higher service levels can be achieved for a given budget with the integrated approach. We gain increasingly significant benefits from integration when inventory decisions become more dominant (i.e., the problems with high inventory holding costs caused by expensive parts, higher inventory levels due to longer time windows, or higher required service levels). This is intuitive since the decoupled approach inherently ignores inventory decisions while designing the network, and, when these decisions become more dominant and interact with network design decisions more, the cost of ignoring them gets higher.

With this study, we show that the benefit of the integrated model can be significant so that the additional computational effort is a worthy investment. In the following chapters, we focus on developing efficient solution methodologies to solve larger problems. We have several alternatives to attack the problem including techniques based on Lagrangian relaxation as discussed in the forth chapter, and column generation as discussed in the fifth chapter. Another possibility is to consider an iterative solution technique based on the

idea of repeating the stages of the decoupled approach in a controlled manner, which also provides additional motivation to analyze the stage submodels independently. We leave this item for the future research.

Chapter 3

Customer-Centric Service Levels

3.1 Introduction

Time-based responsiveness in SPL networks are defined through target service levels. A real SPL system can have mixed types of responsiveness definitions due to the variety of service contracts that a company may have with its customers. For our purposes, however, we define three different problems, depending on the definition of the responsiveness that can be used in an SPL system as a service measure. These eventually determine the specific forms of the service level constraints:

i. *System-Wide (SW) Service Levels*: One extreme is defining service levels for the overall system, such that the percentage of demand satisfied within the time window (including part availability) is greater than the target system-wide service (α). Hence, $100\alpha\%$ of the total demand needs to be satisfied within the time window w. We formulate this problem for multiple parts, and show the benefits of integrated model by comparing it with traditional approach of solving network design and inventory problems seperately (see Chapter 2).

ii. *Regional/Customer-Group (CG) Service Levels*: A more realistic definition would be having target service levels for different groups of customers. Grouping may occur by geographical regions (α_r), (such as customers in New England, or customers in Texas). Alternatively, grouping may be due to different levels of criticality, i.e. customers with high criticality, customers with medium criticality and customers with low criticality. A version of this problem can be solved in a way similar to the SW service level problem if we decompose the overall problem with respect to customer groups, and solve each group subproblem separately.

iii. *Customer-Centric (CC) Service Levels*: Another extreme is defining service levels for each customer, such that the percentage of the customer's demand satisfied within the time window (including part availability) is greater than the target service level of that customer (α_j).

With SW service levels, we deal with demand weighted average service level for the overall system, hence we may end up with some customers having very high (i.e., 100%) service levels, while some others having very low (i.e., 0%) service levels, still satisfying overall system target service level. With smaller likelihood (depending on the size of the customer groups), we may encounter similar consequences in the CG problem for the customers within a group. Because of having service level requirements separately for each customer, the customer-centric service levels do not have the disadvantages of SW or CG service levels mentioned above.

Motivated by these real challenges in today's SPL systems, we model the integrated network design and inventory stocking problem for the problem with customer service levels, where inventory decisions are explicitly considered in the LND problem. We provide two different formulations, each applicable to small and medium size problems. For large problem instances, we provide a relaxation based algorithm which has polynomially solvable subcases.

3.2 Problem Definition and Modeling

3.2.1 Notation and Modeling Assumptions

We are given a set of candidate facility locations I (indexed by i), and a set of demand points J (indexed by j). When we open facility i (or more correctly, locate a facility at candidate location i), we incur a fixed cost of f_i and we incur holding cost of h_i for each unit kept in stock. The unit transportation cost between facility i and demand point j is c_{ij}. Let τ_{ij} be the transportation time from facility i to demand point j. Comparing τ_{ij} to the service time window w, we obtain δ_{ij}, the identifier which takes value 1 if facility i can ship a part requested at demand point j within the specified service time window ($\tau_{ij} \leq w$), 0 otherwise ($\tau_{ij} > w$). The mean demand rate (for a given time unit) is d_j at demand point j.

We make the following assumptions to facilitate model development:

- We assume that network design involves the stocking facilities that are all in one echelon facing the direct demand from geographically dispersed

customers. We assume that these facilities to be located are replenished from a central warehouse with infinite capacity (that is, the central warehouse can replenish the stocking facilities anytime without any delay). The lead times from the central warehouse to all facilities are known and constant.

- Due to the low-demand nature of the motivating SPL problem, we assume that the facilities use continuous review, one-for-one (or base-stock, also called $(S-1, S)$) replenishment policy. This is typical as demands are low, and lead times are relatively short in SPL systems.

- We assume that there is a single part in the overall system.

- We assume that demand at each demand point arrives one at a time according to an independent Poisson process, which is typical in low-demand settings. We further assume that we know the mean demand rates obtained from the part failure rate distributions and the number of parts used at each demand point. Any unsatisfied demand due to a stockout at a facility is backordered. Note that demands (failure rates) for high-cost, critical parts in SPL systems are very low, as these parts are made extremely reliable. See Muckstadt (2005) for standard assumptions in SPL systems.

- We assume that we have one service time window defined. Although typical SPL systems have multiple "tiered" service time windows (with increasing service level requirements for longer time windows), usually

one of the time windows is the most restrictive, hence assuming one window for each part should not hinder the model's value. Besides, modifying the model (as will be seen) for multiple windows is straightforward. In the experiments, however, we vary the time window as a control factor to see its effect on the results.

- We assume that we know which customers a facility can serve within the service time window. As this is usually a function of distance and the mode of transportation available to the facility and customer, we assume that this processing of transportation times is performed for each customer and facility pair *a priori*. We further assume that each customer's part request is satisfied by a single direct shipment from a facility, without any shipment consolidation or bundling. Not only this is an actual practice in SPL systems, but also it is very unlikely to have time or opportunity to consolidate multiple shipments due to low demand and strict time windows.

- We finally assume that it is not possible to split a customer's demand while allocating it to facilities, i.e., demand at a demand point has to be served by a single source.

3.2.2 On the Fill Rate Function

For a facility facing Poisson-distributed demand, the fill rate function is defined as $\beta(s, \lambda) = G(s-1, \lambda)$ for stock level s and mean lead time demand λ, where G is cumulative (Poisson) distribution function, $G(s-1, \lambda) =$

$\sum_{k=0}^{s-1} \lambda^k e^{-\lambda}/k!$. We now analyze a series of properties for the fill rate function, which will be useful to define the customer-centric service level constraints.

Proposition 1. *For a given stock level s, $\beta(s, \lambda)$ is strictly decreasing function of λ, hence assigning more demand to a facility decreases its fill rate.*

Proof. This can be proved by using the first partial derivative of fill rate function with respect to λ as:

$$
\begin{aligned}
\frac{\partial}{\partial \lambda}\beta(s,\lambda) &= \frac{\partial}{\partial \lambda}\sum_{k=0}^{s-1}\frac{\lambda^k e^{-\lambda}}{k!} \\
&= \sum_{k=0}^{s-1} e^{-\lambda}\left(\frac{\lambda^{k-1}}{(k-1)!} - \frac{\lambda^k}{k!}\right) \\
&= \quad e^{-\lambda}(0-1) \\
&\quad + e^{-\lambda}(1-\lambda) \\
&\quad + e^{-\lambda}\left(\lambda - \frac{\lambda^2}{2}\right) \\
&\quad + e^{-\lambda}\left(\frac{\lambda^2}{2} - \frac{\lambda^3}{6}\right) \\
&\quad + \cdots \\
&\quad + e^{-\lambda}\left(\frac{\lambda^{s-3}}{(s-3)!} - \frac{\lambda^{s-2}}{(s-2)!}\right) \\
&\quad + e^{-\lambda}\left(\frac{\lambda^{s-2}}{(s-2)!} - \frac{\lambda^{(s-1)}}{(s-1)!}\right)
\end{aligned}
$$

$$
\begin{aligned}
&= e^{-\lambda}\left(0 - \frac{\lambda^{(s-1)}}{(s-1)!}\right) \\
&= -e^{-\lambda}\frac{\lambda^{(s-1)}}{(s-1)!}.
\end{aligned}
$$

Note that the second term of in each line of the expanded form of the summation is cancelled with the first term of the next line, and we end up with $\frac{\partial}{\partial\lambda}\beta(s,\lambda) = -e^{-\lambda}\frac{\lambda^{(s-1)}}{(s-1)!}$, which is always negative, hence the fill rate is strictly decreasing with respect to λ for given stock level s. □

Proposition 2. *The fill rate is a concave function of λ for a given stock level s, when $\lambda < s-1$, and it is convex when $\lambda \geq s-1$.*

Proof. By proposition 1, $\frac{\partial}{\partial\lambda}\beta(s,\lambda) < 0$ and,

$$
\begin{aligned}
\frac{\partial^2}{\partial\lambda^2}\beta(s,\lambda) &= \frac{\partial^2}{\partial\lambda^2}\left(-e^{-\lambda}\frac{\lambda^{(s-1)}}{(s-1)!}\right) \\
&= \frac{\lambda^{s-2}e^{-\lambda}}{(s-1)!}(\lambda+1-s).
\end{aligned}
$$

From this, we note that

- $\frac{\partial^2}{\partial\lambda^2}\beta(s,\lambda) < 0$ for $\lambda < s-1$, hence the fill rate function is concave for $\lambda < s-1$
- $\frac{\partial^2}{\partial\lambda^2}\beta(s,\lambda) \geq 0$ for $\lambda \geq s-1$, hence the fill rate function is convex for $\lambda \geq s-1$.

□

Figure 3.2 illustrates fill rate as a function of mean lead time demand λ for stock levels 1 and 2. Note that, when $s = 1$, the fill rate function is convex for any mean lead time demand ($\lambda \geq 0$), and when $s = 2$, the fill rate function is concave for $\lambda < 1$ and convex for $\lambda \geq 1$.

Suppose now that we need to guarantee a fill rate level of b at a facility with stock level s facing mean lead time demand of λ. We define parameter $\mu(s)$ as the maximum mean lead time demand that can be assigned to this facility to guarantee a fill rate level of b or higher. $\mu(s)$ can be solved from the equation

$$b = \sum_{k=0}^{s-1} \frac{\mu(s)^k e^{-\mu(s)}}{k!}. \tag{3.1}$$

Although $\mu(0)$ and $\mu(1)$ can be solved explicitly from equation (3.1) as a closed form function of other parameters, $\mu(s)$ for $s \geq 2$ does not have a closed-form equation. However, they can be calculated numerically in mathematical software. More specifically, the fill rate as a function of stock level s and mean lead time demand λ, $\beta(s, \lambda) = Q(s, \lambda)$ is the incomplete regularized gamma function with the same arguments, where $Q(s, \lambda) = \frac{\Gamma(s,\lambda)}{\Gamma(s)}$, $\Gamma(s, \lambda) = \sum_{r=0}^{s-1} \lambda^r e^{-\lambda}$ is the incomplete gamma function, and $\Gamma(s)$ is the complete gamma function which is $\Gamma(s) = (s-1)!$ since s is integer. Hence, we calculate $\mu(s)$ parameters for given fill rates and stock levels numerically, using the inverse of the regularized gamma function in Mathematica.

From the numerical analysis, we conjecture that for $\beta \geq 0.5$, $\mu(s + 2) - \mu(s + 1) \geq \mu(s + 1) - \mu(s)$ which states, to guarantee a fixed fill rate b,

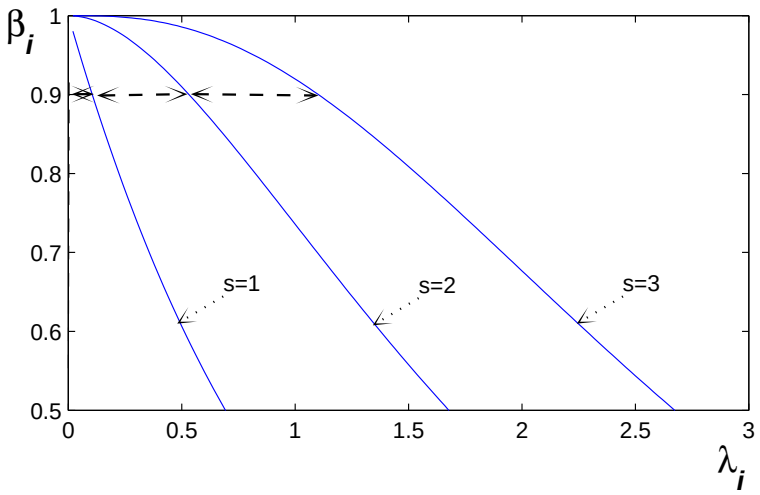


Figure 3.1: Fill rates for stock levels $s = 1$, 2, and 3

the maximum lead time demand that can be assigned to a facility increases more for higher stock levels than for smaller stock levels when stock level is increased by 1 unit. That is, the additional mean demand that a facility can be assigned increases as the stock level increases as shown in Figure 3.1. In other words, $\mu(s)$ is a discretely convex function of s, i.e., for 90% fill rate, $\mu(1) - \mu(0) = 0.11$, $\mu(2) - \mu(1) = 0.43$, and $\mu(3) - \mu(2) = 0.57$. With this conjecture we simplify our formulation in the later analysis by showing that some of the limiting constraints for the stock levels at the facilities are redundant.

3.2.3 Mathematical Formulation

In this section we introduce different mathematical formulations for the customer-centric problem (CCP), starting with a formulation analogous to the single part version of system-wide problem (SWP) defined in the second chapter. The main modification to the model is to rewrite the service level constraints for each customer. The modified formulation is as follows:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} d_j X_{ij} + \sum_{i \in I} h_i S_i \tag{3.2}$$

$$\text{subject to} \quad \sum_{i \in I} X_{ij} = 1, \ \forall j \tag{3.3}$$

$$X_{ij} \leq Y_i \ \forall i, j \tag{3.4}$$

$$S_i \leq L Y_i, \ \forall i \tag{3.5}$$

$$\lambda_i = t_i \sum_{j \in J} d_j X_{ij}, \ \forall i \tag{3.6}$$

$$\beta_i = \sum_{r=0}^{S_i - 1} \lambda_i^r \frac{e^{-\lambda_i}}{r!}, \ \forall i \tag{3.7}$$

$$\sum_{i \in I} \beta_i \delta_{ij} X_{ij} \geq \alpha_j, \ \forall j \tag{3.8}$$

$$X_{ij} = 0 \text{ or } 1, \ \forall i, j \tag{3.9}$$

$$Y_i = 0 \text{ or } 1, \ \forall i \tag{3.10}$$

$$S_i \in \{0, 1, 2, \ldots, L\}, \ \forall i. \tag{3.11}$$

The first term in the objective is the fixed facility cost f_i associated with open/close decision Y_i, the second term is the transportation cost for assign-

ments X_{ij}, and the third term is the inventory stocking costs, which applies unit holding cost h_i and to stock level S_i.

To facilitate the model, we assume that there is a finite number of alternatives we consider for stock levels S_i. We denote its largest element by L, which can be calculated by considering maximum possible demand for a facility. Hence, we consider $S_i \in \{0, 1, 2, \ldots, L\}$ for all i (for brevity, we will use L for both largest element of this set, and also for the name of the set). We make this set large enough to give practically $\bar{\alpha}\%$ fill rate, where $\bar{\alpha} = max_{j \in J} \alpha_j$, at the largest stock level L for total demand that could be assigned to a facility. Hence, constraints (3.11) allow the stock levels to be selected from the initially developed set of integer stock levels.

With this in mind, constraints (3.3) guarantee all demand at each demand points to be fully assigned to some facility. Constraints (3.4) mean that any facility serving a demand point should be open. Constraints (3.5) allow stock levels to be greater than 0 only for open facilities. Constraints (3.8) are the time-based service coverage constraints, each for a demand point, where β_i (calculated in constraints (3.7)) is the fill rate at location i at its stock level S_i and at its mean lead time demand λ_i (calculated in constraints (3.6)), where t_i is the lead time for facility i. Here, the long run fill rate β_i (percentage of demand satisfied directly from inventory on hand) is computed assuming that demand from each customer has an independent Poisson distribution (with mean d_j at customer j) and the inventory policy is one-for-one replenishment (also known as the base-stock policy). In this case, the lead time demand is

Poisson with mean λ_i for location i. (Poisson distribution is a very common assumption for SPL systems, as the demand comes one at a time, and is very low. The base stock inventory policy is also very common, both in practice of SPL and in the literature dealing with low-demand parts, see Muckstadt (2005)).

The model (3.2)-(3.11) is a mixed integer nonlinear programming problem. The main source of nonlinearity is the service level constraints (3.8), since each incorporates the product of a demand allocation variable X_{ij} and the associated fill rate variable β_i, where β_i is a nonlinear function of two other decision variables, stock levels S_i and mean lead time demands λ_i.

Note that, in CCP, customer j has to be served with the promised "positive" service level $\alpha_j > 0$ as spelled out in the customer's contract. Therefore, different than the SWP formulation in the second chapter, in this setting we do not consider assignments of the customers to the facilities out of the time window (because, by definition, the achieved service level is zero for a customer out of the time window). For simplicity, we define set of facilities I_j for each customer j, as facilities that can serve customer j within the time window, $I_j = \{i \in I : \delta_{ij} = 1\}$, and similarly we define set of customers J_i for each facility i, as customers that can be served by facility i within the time window, $J_i = \{j \in J : \delta_{ij} = 1\}$. Here assignment variables X_{ij} are defined only for $j \in J$ and $i \in I_j$. With this simplification, we can rewrite the service level constraint (3.8) as $\sum_{i \in I_j} \beta_i X_{ij} \geq \alpha_j$.

The following proposition helps us simplify the overall model further:

Proposition 3. *In CCP, each customer has to be served by a facility having fill rate higher than the customer's target service level. If facility i serves the customers in set $\hat{J}_i$ such that $\hat{J}_i = \{j \in J_i : X_{ij} = 1\}$, then the fill rate of facility i must be greater than or equal to the highest service level requirement within the set $\hat{J}_i$ such that $\beta_i \geq \max_{j \in \hat{J}_i}\{\alpha_j\}$.*

Note that, the condition in Proposition 3 is sufficient, but may not be necessary if prioritization of critical customers is allowed (see Appendix for an example case). Since we assume that customer requests are satified without any prioritization in a "first-come first-serve" bases, we have the sufficient condition of $\beta_i \geq \max_{j \in \hat{J}_i}\{\alpha_j\}$. Briefly, Proposition 3 states that we should choose the minimum inventory level S_i^* giving the fill rate higher than the maximum target service level of the served customers ($\min_{S_i \in L}\beta_i \geq \max_{j \in \hat{J}_i}\{\alpha_j\}$). To handle the right-hand-side of this condition (choosing the maximum service level of the served customers), we can simply write it for each facility i as

$$\beta_i \geq \alpha_j X_{ij}, \quad \forall j \in J_i. \tag{3.12}$$

Note that constraints (3.12) have two functions: (1) deciding the required fill rate at facility i, and (2) deciding the inventory level at the facility to satisfy this fill rate. To simplify this constraint further, we handle decisions of fill rate and inventory levels in two constraints instead of a single constraint. For example, if we know the fill rate of a facility, $\hat{\beta}_i$, then finding the inventory level for this fill rate is easier.

To find the minimum required fill rates, called *"base fill rates"*, at the facilities, we define a set B_i of potential fill rates for each facility i. If there are N_i (for brevity we use N_i for both largest element of this set, and also for the name of the set) distinct service level requirements in set J_i (usually N_i is much smaller than $|J_i|$), then $B_i = \{b_{i1}, b_{i2}, \ldots, b_{i,N_i}\}$ is the set of distinct service level requirements of the customers in J_i (without loss of generality we assume that b_{in}'s are in increasing order in B_i). We now define binary variable $W_{in}, \forall i, n$, which takes value 1 if the base fill rate at the facility i is equal to b_{in}. To guarantee that the facilities have fill rates greater than or equal to the service level of their served customers, we define the constraints:

$$\sum_{n \in N_i:\, b_{in} \geq \alpha_j} W_{in} \geq X_{ij}, \quad \forall j \in J, i \in I_j$$

(Note that, alternatively this constraint can be written as $\sum_{n \in N_i} b_{in} W_{in} \geq \alpha_j X_{ij}, \quad \forall j \in J, i \in I_j$).

To facilitate the second function of the service level constraints (deciding the stock levels), we calculate the parameters μ_{ins} for all $n \in N_i$ and $s \in L$ from the fill rate function such that $b_{in} = \sum_{k=0}^{s-1} \left((\mu_{ins})^k (e)^{-\mu_{ins}} \right) / k!$. For an illustration of the calculation of μ_{ins}, see Figure 3.2. According to this figure, to guarantee at least 70% fill rate ($b_{i,n} = 0.7$ with $n = 2$), with 1 unit of stock, facility i can accept at most $\mu_{i,n=2,s=1} = 0.36$ units of mean lead time demand ($t_i \sum_{j \in J_i} d_j X_{ij}$), and with 2 units of stock, facility i can accept up to $\mu_{i,s=2,n=2} = 1.1$ units of mean lead time demand. We now define binary variable $Q_{is}, \forall i, s$, which takes value 1 if the stock level at facility i is equal

to s. Then, the following constraint guarantees required stock level s to be selected:

$$t_i \sum_{j \in J_i} d_j X_{ij} \leq \sum_{s \in L} \mu_{ins} Q_{is} + M_i(1 - W_{in}), \quad \forall i \in I, n \in 1..N_i$$

where $M_i = \sum_{j \in J_i} d_j$. Note that, if the base fill rate at facility i is $b_{i\hat{n}}$ (hence $W_{i\hat{n}} = 1$), the constraint will remain as $t_i \sum_{j \in J_i} d_j X_{ij} \leq \sum_{s \in L} \mu_{i\hat{n}s} Q_{is}$, which is basically a forcing constraint for facility i. We now write the complete, linearized and simplified model:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J_i} c_{ij} d_j X_{ij} + \sum_{i \in I} h_i \left(\sum_{s \in L_i} s Q_{is} \right) \tag{3.13}$$

$$\text{subject to} \quad \sum_{i \in I_j} X_{ij} = 1, \ \ \forall j \in J \tag{3.14}$$

$$X_{ij} \leq Y_i, \ \ \forall i \in I_j, j \in J \tag{3.15}$$

$$X_{ij} \leq \sum_{n \in N_i(\alpha_j)} W_{in}, \ \ \forall j \in J, i \in I_j \tag{3.16}$$

$$t_i \sum_{j \in J_i} d_j X_{ij} \leq \sum_{s \in L_{in}} \mu_{ins} Q_{is} + M_i(1 - W_{in}), \ \ \forall i \in I, n \in N_i \tag{3.17}$$

$$\sum_{s \in L_i} Q_{is} \leq 1, \ \ \forall i \in I \tag{3.18}$$

$$\sum_{n \in N_i} W_{in} \leq 1, \ \ \forall i \in I \tag{3.19}$$

$$X_{ij} = 0 \text{ or } 1, \ \ \forall i \in I, j \in J_i \tag{3.20}$$

$$W_{in} = 0 \text{ or } 1, \ \ \forall i \in I, n \in N_i \tag{3.21}$$

$$Q_{is} = 0 \text{ or } 1, \ \ \forall i \in I, s \in L_i. \tag{3.22}$$

Here, the first term in the objective (3.13) is a fixed facility cost f_i, the second term is the transportation cost for assignments X_{ij}, and the third term

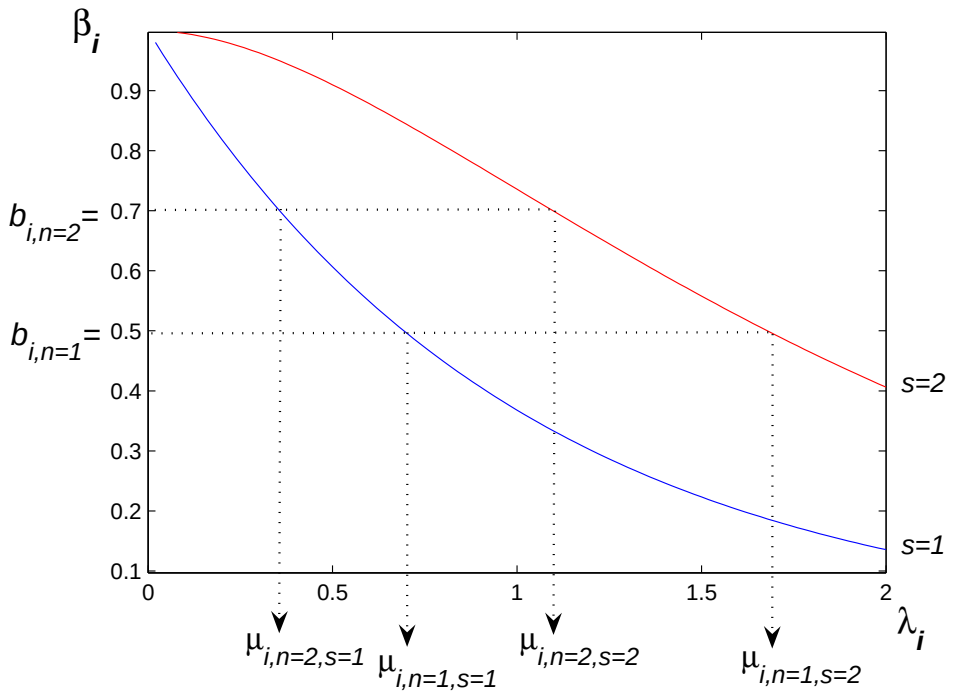


Figure 3.2: Calculation of μ_{ins} assuming two distinct service levels ($N_i = 2$) in customer set J_i, where $B_i = \{b_{i1} = 0.5, b_{i2} = 0.7\}$ for stock levels $L_i = \{1, 2\}$

is the inventory stocking costs captured with stock choice (binary) variables Q_{is}, that iterates through the available stock levels for each facility i (note that $\sum_{s \in L_i} sQ_{is}$ gives the stock level at facility i). Constraints (3.14) assign all the demands to a facility within the time window while constraints (3.16)-(3.17) guarantee the service level requirements where $N_i(\alpha_j) = \{n \in N_i : b_{in} \geq \alpha_j\}$.

Note that in the formulation above, we are using different maximum potential inventory levels L_{in} for each facility i and the n^{th} base fill rate level b_{in}. We could simply define a common L for all i and n by considering all of the

customers' total lead time demands, say $\Lambda = t \sum_{j \in J} d_j$ where t is the longest lead time ($t = \max_{i \in I}\{t_i\}$), and calculating L for a very high fill rate value say 99.9% by solving for L in $\sum_{k=0}^{L-1} \left(\Lambda^k e^{-\Lambda}\right)/k! \geq 0.999$. However, the problem size increases with L. Hence we use the tightest value, depending on the facility (since each facility may cover different amount of total demand within its time window) and on the fill rate level (since the required inventory level increases with fill rate requirements). Therefore, we calculate L_{in} by solving for it in $\sum_{k=0}^{L_{in}-1} \left(\lambda_i^k e^{-\lambda_i}\right)/k! \geq b_{in}$, where $\lambda_i = t_i \sum_{j \in J_i} d_j$. Using Figure 3.2 as an illustration, if the total mean lead time demand within facility i's time window coverage is 0.5 units ($\lambda_i = 0.5$), to guarantee a $b_{in} = 50\%$ fill rate, facility i may need 1 unit of stock. However, to guarantee a $b_{in} = 70\%$ fill rate, facility i may need up to 2 units of stock. Hence, for the example in Figure 3.2, $L_{in} = 1$ for $n = 1$ and $L_{in} = 2$ for $n = 2$. We use the notation L_i for the maximum of L_{in}'s for facility i (for this example $L_i = 2$).

Proposition 4. *Constraints (3.19) are redundant.*

Proof. Because of constraints (3.17), having multiple W_{in} variables equal to 1 forces an inventory level to be chosen for each of them. However, constraints (3.18) do not allow this, and hence there is no explicit need for constraints (3.19). □

Proposition 5. *Assuming that the conjecture we made in Section 3.2.2 is true, constraint (3.18) is redundant for facility i (If $\alpha_j \geq 0.5$ for all $j \in J_i$).*

Proof. Let us assume that for facility i, $\sum_{s \in L_i} Q_{is} \geq 1$, where $Q_{is_1} = 1$ and $Q_{is_2} = 1$. Then, inventory stocking cost at facility i is $h_i(s_1 + s_2)$. By constraints (3.17), facility i can serve mean lead time demand up to $\mu_{ins_1} + \mu_{ins_2}$.

With same inventory stocking cost (ignoring the fixed cost of opening facility i twice), facility i may be open with stock level of $s_3 = s_1 + s_2$, and can serve lead time demand up to μ_{ins_3}. By the conjecture we made in Section 3.2.2, we know that the additional mean demand that a facility can be assigned increases as the stock level increases, hence $\mu_{ins_3} \geq \mu_{ins_1} + \mu_{ins_2}$. Therefore, opening facility i with s_3 is preferable to opening it with s_1 and s_2, hence we do not need constraints (3.18). □

As an example for Proposition 5, for a 90% fill rate, if facility i is open with stock levels $s_1 = 1$ and $s_2 = 1$ having stocking cost of $2h_i$, the maximum mean lead time demand that can be assigned to facility i is $\mu_{i,s_1,n} + \mu_{i,s_2,n} = 0.11 + 0.11$. Whereas, if facility i is open with stock level $s_3 = 2$ having same stocking cost, it can be assigned $\mu_{i,s_3,n} = 0.54$ unit mean lead time demand.

With this formulation (which we call formulation $F1$ as we introduce a strengthened version called $F2$ later), we can solve small and medium size problems (i.e., up to 30 facilities and 100 customers), but it is still very hard to solve larger problems. From our experiments, we see that $F1$ has very loose LP relaxation bounds, resulting in excessive solution times to close the optimality gap. To improve this formulation, we seek to tighten the formulation, by adding cuts and/or having stronger constraints with higher dimensions if

possible. In the following section, we propose an alternative formulation, denoted by $F2$ with a tighter solution space, and then compare it with $F1$ in terms of solution times and initial LP relaxation bounds.

3.2.4 Improved Formulation

Instead of using separate binary variables Y_i, W_{in} and Q_{is} for choosing facilities, inventory levels and stock levels, one may define composite variables Z_{ins} which is 1 when facility i is open with fill rate b_{in} and stock level s, and 0 otherwise. We also expand assignment variables X_{ij} to X_{insj}. The idea behind this notation is to consider each (i, n, s) as a node with predefined fill rate and stock level that a customer may be assigned.

The formulation with composite and expanded variables are known to have stronger relaxation bounds (see Barnhart and Cohn (2004) for an example of such use of composite variables). With the modification described, the new formulation is as follows:

$$\min \sum_{i\in I} f_i \left(\sum_{n\in N_i} \sum_{s\in L_i} Z_{ins} \right) + \sum_{i\in I} \sum_{n\in N_i} \sum_{s\in L_i} \sum_{j\in J_i} c_{ij} d_j X_{insj} + \sum_{i\in I} h_i \left(\sum_{n\in N_i} \sum_{s\in L_i} s Z_{ins} \right) \tag{3.23}$$

subject to

$$\sum_{i\in I_j} \sum_{n\in N_i(\alpha_j)} \sum_{s\in L_i} X_{insj} = 1, \quad \forall j \in J \tag{3.24}$$

$$Z_{ins} \geq X_{insj}, \quad \forall j \in J, i \in I_j, n \in N_i(\alpha_j), s \in L_i \tag{3.25}$$

$$t_i \sum_{j\in J_i} d_j X_{insj} \leq \mu_{ins}, \quad \forall i \in I, n \in N_i, s \in L_i \tag{3.26}$$

$$\sum_{n\in N_i} \sum_{s\in L_i} Z_{ins} \leq 1, \quad \forall i \in I \tag{3.27}$$

$$X_{insj} = 0 \text{ or } 1, \quad \forall i \in I, n \in N_i, s \in L_i, j \in J_i. \tag{3.28}$$

Note that we can combine the first and last terms of the objective function as

$$\sum_{i\in I} \sum_{n\in N_i} \sum_{s\in L_i} (f_i + h_i s) Z_{ins}.$$

The constraints are analogous to the previous model with new demand satisfaction constraints (3.24), fill rate constraints (3.25), and inventory constraints (3.26). We note that using composite variables especially simplifies the inventory constraints. Since binary assignment variables X_{insj} are forcing each W_{ins}, we can relax binary constraints for W_{ins}.

Proposition 6. *Constraints (3.27) are redundant. Then (3.23)-(3.26) along with (3.28) is the capacitated facility location problem, with facilities defined for each combination (i, n, s), fixed costs $(f_i + h_i s)$, and capacities μ_{ins}/t_i.*

Although we do not provide a formal proof of equivalncy of two formulations $F1$ and $F2$, or we do not show $F2$ is stronger formulation of the same problem, we belive that $F1$ is weaker due to use of "big-M" and more aggregate variables.

3.2.5 Comparisons

Although the model with system-wide service level is highly nonlinear and out of reach of our current solution technologies even after linearization (for reasonable size problems), the counter part with customer-centric service levels leads to formulations easier in terms of linearity and decomposition. For this problem we now have two competing formulations, $F1$ and $F2$.

In this section we discuss advantages and disadvantages of these formulations and compare them in terms of their problem sizes, solution times, and quality of the LP relaxations with a set of experimental results. Then, we select one of the formulations as the basis for methodology development.

Formulation (3.13)-(3.22) for CCP, $F1$, has $|I|(1 + |J| + |N| + |S|)$ variables and $|I|(3 + 2|J| + |N|)$ constraints, whereas in the new formulation, $F2$, there are $|I||J||N||S|$ binary and $|I||N||S|$ continuous variables, with $|I| + |J| + |I||N||S|(1 + |S|)$ constraints. For the medium size instance we use in our experiments (with 50 facilities and 200 customers), $F1$ has about 11,000 variables and 20,000 constraints, whereas $F2$ has about 1,000,000 variables and 300,000 constraints. The main reason for $F2$ formulation's size to be much larger than $F1$ is having assignment variables for all (i, n, s, j) combinations.

Table 3.1: Experimental data set

Data	a, b
Size ($\|I\|$, $\|J\|$)	small (25, 100), medium (50, 200), large (100, 400)
Mean demand, d	low [1, 4], high [4, 8]
Time window, w	2 hours, 4 hours
Holding cost, h	\$250, \$500, \$1000
Fixed cost, f	\$1000
Transportation cost, c	\$5 per unit distance
Total	72 instances (named as a1-a36 and b1-b36)

Table 3.1 summarizes the experimental data set we use to compare $F1$ and $F2$. We generate two different data sets, denoted by a and b, with different size instances (a small size instance with $|I| = 25$, $|J| = 100$, a medium size instance with $|I| = 50$, $|J| = 200$, and a rather large instance with $|I| = 100$, $|J| = 400$). Mean demand rates are generated uniformly, with a mean of 2.5 (continuous uniform between 1 and 4) for low demand instances, and a mean of 6 (continuous uniform between 4 and 8) for high demand instances. We have two time windows, 2 hours and 4 hours. Each instance is run with low (\$250), medium (\$500) and high (\$1000) holding costs. Fixed facility costs are \$1000 and transportation costs are distance-based and average transportation cost is around \$150. We randomly split customers into 3 groups, requiring high (90%), medium (70%) and low (50%) target service levels. We limit the maximum solution time to 3600 seconds.

Tables 3.2 and 3.3 summarize the results of the experiments for data sets a and b respectively. We provide initial LP relaxation solution values, best

feasible solution values (which is optimal if completed within 3600 seconds), the solution times and the status of the final solutions obtained (where N/F means "not finished", and Opt means optimal is found). We also provide final gaps between lower and upper bound calculated as $100 * (BestUB - FinalLB)/BestUB$, which is zero when an optimal solution is found.

We solve these problems with Xpress-MP by Dash Optimization. Note that, Xpress-MP uses presolve and cut generation techniques together with branch-and-bound, which have significant effect on the solution time (e.g., with formulation 2, the instance a13 is solved to optimality within 36 seconds, whereas it has more than 2% gap at the end of 3600 seconds when cut generation is disabled).

Table 3.2: Comparison of $F1$ and $F2$ with data a

					Formulation 1						Formulation 2					
ins	size	d	w (hrs)	h ($)	LP soln	Time	Best UB	Time	Status	Gap	LP soln	Time	Best UB	Time	Status	Gap
a1	small	low	2	250	15790	0.1	30210	3600	N/F	2.6	28848	0.4	30210	1	Opt	0.0
a2				500	16766	0.0	35460	3600	N/F	3.1	32793	0.3	35460	2	Opt	0.0
a3				1000	18507	0.0	46107	3600	N/F	6.0	40233	0.3	45960	4	Opt	0.0
a4			4	250	12446	0.1	29992	3600	N/F	25.2	27217	2.7	28919	18	Opt	0.0
a5				500	12446	0.1	33801	3600	N/F	30.8	29551	3.6	32889	53	Opt	0.0
a6				1000	12446	0.1	43242	3600	N/F	43.9	33455	5.9	39906	57	Opt	0.0
a7		high	2	250	34392	0.0	52962	3600	N/F	4.8	50318	0.4	52775	1	Opt	0.0
a8				500	35774	0.0	61611	3600	N/F	6.8	56246	0.4	61275	6	Opt	0.0
a9				1000	38124	0.0	78867	3600	N/F	10.8	67781	0.4	78016	163	Opt	0.0
a10			4	250	30214	0.1	55735	3600	N/F	16.7	50205	2.8	52775	11	Opt	0.0
a11				500	30214	0.1	63343	3600	N/F	19.0	55592	5.0	61079	604	Opt	0.0
a12				1000	30214	0.1	81619	3600	N/F	32.4	64499	6.9	75628	2769	Opt	0.0
a13	medium	low	2	250	28371	0.1	56782	3600	N/F	21.2	50650	2.7	53499	36	Opt	0.0
a14				500	30213	0.1	70984	3600	N/F	30.7	56515	3.0	62749	1004	Opt	0.0
a15				1000	33454	0.1	90660	3600	N/F	35.0	67558	3.8	80723	3644	N/F	1.2
a16			4	250	22004	0.3	56655	3600	N/F	40.2	47707	45.2	50602	1252	Opt	0.0
a17				500	22004	0.3	68430	3600	N/F	50.4	51535	71.2	57939	3620	N/F	0.3
a18				1000	22004	0.3	84328	3600	N/F	59.0	57504	119.5	74209	3615	N/F	4.4
a19		high	2	250	59058	0.1	102804	3600	N/F	19.3	91249	2.5	96196	45	Opt	0.0
a20				500	62157	0.1	119603	3600	N/F	23.0	100756	3.2	111332	3600	N/F	0.5
a21				1000	66603	0.1	151092	3600	N/F	31.0	117951	3.8	142984	3600	N/F	3.2
a22			4	250	51892	0.3	109947	3600	N/F	32.8	90008	70.5	95116	968	Opt	0.0
a23				500	51892	0.3	126925	3600	N/F	39.4	98459	98.8	111791	3600	N/F	2.4
a24				1000	51892	0.3	158591	3600	N/F	51.2	111211	185.9	137158	3600	N/F	1.5
a25	large	low	2	250	31491	0.5	104797	3600	N/F	54.2	75499	105.4	82899	3600	N/F	1.5
a26				500	31671	0.5	122430	3600	N/F	59.5	82276	164.5	118778	3600	N/F	20.2
a27				1000	31769	0.6	167634	3600	N/F	69.6	93788	242.2	154298	3600	N/F	22.4
a28			4	250	29860	1.4	120452	3600	N/F	63.0	75127	1587.8	199372	3600	N/F	60.3
a29				500	29860	1.4	149273	3600	N/F	69.9	81173	2392.6	245070	3600	N/F	64.9
a30				1000	28985	1.4	203801	3600	N/F	79.6	89241	3425.2		3600	N/F	
a31		high	2	250	73157	0.6	183793	3600	N/F	42.9	138954	82.7	150592	3600	N/F	0.6
a32				500	73664	0.6	223200	3600	N/F	52.2	153504	185.0	187268	3600	N/F	6.0
a33				1000	74439	0.6	283862	3600	N/F	61.2	175875	331.3	332097	3600	N/F	34.4
a34			4	250	69902	1.4	195209	3600	N/F	46.5	138954	1801.4		3600	N/F	
a35				500	69902	1.4	234224	3600	N/F	55.6				3600	N/F	
a36				1000	69902	1.4	346458	3600	N/F	69.8				3600	N/F	

Table 3.3: Comparison of $F1$ and $F2$ with data b

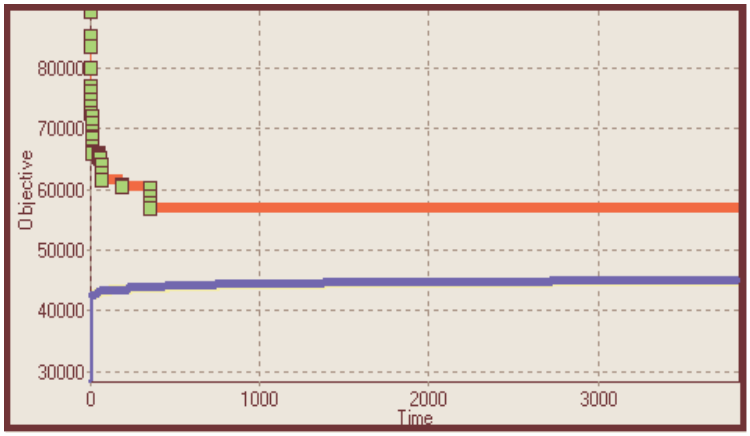
					Formulation 1						Formulation 2					
ins	size	d	w (hrs)	h ($)	LP soln	Time	Best UB	Time	Status	Gap	LP soln	Time	Best UB	Time	Status	Gap
b1	small	low	2	250	13931	0.0	31983	3600	N/F	5.3	30073	0.4	31983	3	Opt	0.0
b2				500	14294	0.0	37521	3600	N/F	6.6	33853	0.3	37521	2	Opt	0.0
b3				1000	14747	0.0	49400	3600	N/F	12.7	41268	0.3	48521	4	Opt	0.0
b4			4	250	11874	0.1	29582	3600	N/F	20.6	27405	1.1	28881	8	Opt	0.0
b5				500	11874	0.1	35402	3600	N/F	29.8	29846	1.6	32656	9	Opt	0.0
b6				1000	11874	0.0	42476	3600	N/F	33.1	33775	2.2	39686	25	Opt	0.0
b7		high	2	250	31997	0.0	54905	3600	N/F	7.3	52149	0.5	54629	2	Opt	0.0
b8				500	32710	0.0	64159	3600	N/F	9.7	58337	0.4	63415	15	Opt	0.0
b9				1000	33689	0.0	82019	3600	N/F	14.2	69938	0.4	80467	238	Opt	0.0
b10			4	250	28832	0.1	55646	3600	N/F	14.6	51104	1.9	53411	27	Opt	0.0
b11				500	28832	0.0	64794	3600	N/F	21.7	56419	3.4	61369	281	Opt	0.0
b12				1000	28832	0.1	82011	3600	N/F	32.7	64260	3.6	76269	3600	N/F	0.4
b13	medium	low	2	250	21953	0.1	55881	3600	N/F	23.9	48232	2.5	51519	1894	Opt	0.0
b14				500	21992	0.1	69385	3600	N/F	33.6	53733	3.1	59807	1367	Opt	0.0
b15				1000	22063	0.1	94409	3600	N/F	43.7	63913	3.4	76830	3600	N/F	1.4
b16			4	250	20989	0.3	57533	3600	N/F	42.6	46220	45.6	48994	360	Opt	0.0
b17				500	20989	0.3	69132	3600	N/F	51.6	50496	57.5	56520	3600	N/F	0.4
b18				1000	20989	0.3	86882	3600	N/F	60.9	57102	118.2	73142	3600	N/F	4.6
b19		high	2	250	49385	0.1	100573	3600	N/F	22.3	85871	2.4	90890	190	Opt	0.0
b20				500	49448	0.1	120554	3600	N/F	28.9	95181	3.4	105503	2394	Opt	0.0
b21				1000	49506	0.1	152238	3600	N/F	35.4	111608	5.3	136857	3600	N/F	2.7
b22			4	250	47299	0.3	104764	3600	N/F	30.4	85769	51.0	90884	2409	Opt	0.0
b23				500	47299	0.3	128708	3600	N/F	41.3	94272	97.5	105705	3600	N/F	0.6
b24				1000	47491	0.3	155684	3600	N/F	53.0	105628	209.2	134057	3600	N/F	3.1
b25	large	low	2	250	32311	0.6	103519	3600	N/F	54.4	75281	90.2	81872	3600	N/F	0.5
b26				500	32311	0.5	130165	3600	N/F	63.1	81615	125.8	134395	3600	N/F	29.7
b27				1000	32311	0.5	166456	3600	N/F	69.5	92554	190.7	179903	3600	N/F	33.8
b28			4	250	31599	1.4	118857	3600	N/F	62.1	75095	1248.8	151397	3600	N/F	47.0
b29				500	31599	1.4	154471	3600	N/F	70.9	81268	1885.7	244420	3600	N/F	64.3
b30				1000	31599	1.4	206096	3600	N/F	78.1	91327	2935.0	315842	3600	N/F	
b31		high	2	250	75791	0.5	184609	3600	N/F	43.5	139622	92.1	154180	3600	N/F	2.4
b32				500	75791	0.6	221796	3600	N/F	52.2	152793	174.4	181316	3600	N/F	3.0
b33				1000	75791	0.5	282311	3600	N/F	60.6	173579	271.4	305612	3600	N/F	28.4
b34			4	250	74116	1.3	208510	3600	N/F	49.9	139622	1968.2		3600	N/F	
b35				500	74116	1.3	251581	3600	N/F	58.9	152792	3246.6		3600	N/F	
b36				1000	74116	1.3	354921	3600	N/F	70.7				3600	N/F	

As shown in Table 3.2, $F1$ finds LP relaxation solutions very quickly (less than 2 seconds in all of the instances), however they provide bounds that are very loose. Meanwhile, the average LP solution time of $F2$ is much longer, and $F2$ can not even find the LP relaxation solution within 3600 seconds for some large instances (a35, a36 and b36). On the other hand, the average improvement of $F2$ LP bounds (when it has one) over $F1$ bounds is about 120%, ranging from 46% (for instance a7) to 210% (for instance a30).

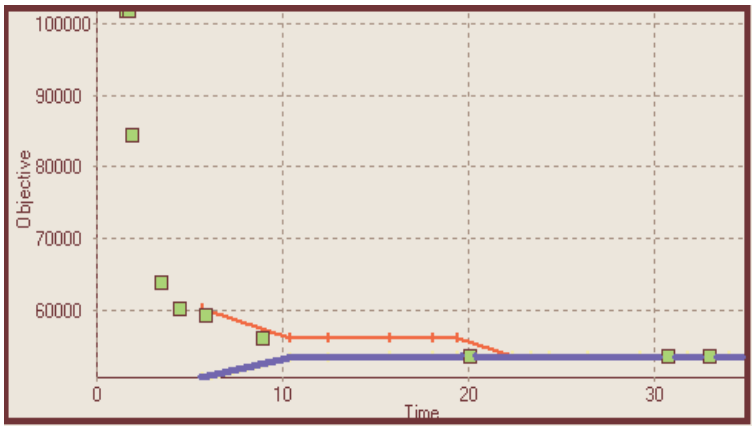
We see that $F1$ can not finish any instance to optimality. Actually, $F1$ solutions for four instances are optimal (a1, a2, and b1, b2), but they are not proven optimal due to the gap. On the other hand, $F2$ finishes almost all of the small and some of the medium size instances to optimality within 3600 seconds (average improvement over $F1$ upper bound is 17% for these instances). However, for the large instances $F2$ looses its superiority over $F1$, and for 16 of the large instances (a28-a30, a33-a36, and b26-b30, b33-b36) $F1$ upper bounds are better.

To better investigate the performance of both formulations, we show their lower and upper bounds, for a medium instance (a13) in Figure 3.3. For this instance, $F1$ has 1340 variables and 1452 constraints, whereas $F2$ has 6172 variables and 6371 constraints. The squares show the feasible solutions, and lines show lower and upper bounds. $F2$ finds an optimal solution in about 30 seconds (see Figure 3.3.ii), but $F1$ has more than 20% gap between the lower and upper bounds (see Figure 3.3.i).

We conclude that formulation $F2$ has faster convergence and better



i. F1



ii. F2

Figure 3.3: Lower and upper bound plots of F1 and F2 for instance a13

quality lower bounds in general, but it struggles as problem size increases. Note that, for the largest problem (a36), $F1$ has about 5000 variables and constraints, but $F2$ has more than 70000 variables and constraints. In addition, $F2$ also has structural advantages for decomposition. Therefore, we propose $F2$ for solving small and medium size problems and for using as a

base of our methodology development to solve larger instances.

3.3 Methodology

In this section, we use the following simplified version of formulation F2, where $f_{is} = f_i + h_i s$ is the fixed cost of opening facility i with stock level s and $\bar{\mu}_{ins} = \mu_{ins}/t_i$ is maximum annual mean demand that can be served by facility i with fill rate of b_{in} and stock level of s. We propose different lower bounding techniques based on relaxation and decomposition in Section 3.3.1. Then, we develop an upper bound algorithm and variable fixing scheme in Section 3.3.2. Finally, we combine these techniques in a subgradient algorithm to obtain solutions potentially close to optimum for large problems.

For completeness, we have the following model:

$$\min \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i} f_{is}Z_{ins} + \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i}\sum_{j\in J_i} c_{ij}d_jX_{insj} \tag{3.29}$$

$$\text{subject to} \quad \sum_{i\in I_j}\sum_{n\in N_i(\alpha_j)}\sum_{s\in L_i} X_{insj} = 1 \;\; \forall j \in J \tag{3.30}$$

$$Z_{ins} \geq X_{insj} \;\; \forall j \in J, i \in I_j, n \in N_i(\alpha_j), s \in L_i \tag{3.31}$$

$$\sum_{j\in J_i} d_jX_{insj} \leq \bar{\mu}_{ins} \;\; \forall i \in I, n \in N_i, s \in L_i \tag{3.32}$$

$$\sum_{n\in N_i}\sum_{s\in L_i} Z_{ins} \leq 1 \;\; \forall i \in I \tag{3.33}$$

$$X_{insj} = 0 \text{ or } 1, \;\; \forall i \in I, n \in N_i, s \in L_i, j \in J_i \tag{3.34}$$

$$Z_{ins} = 0 \text{ or } 1, \;\; \forall i \in I, n \in N_i, s \in L_i. \tag{3.35}$$

3.3.1 Lower Bound

In this section, we first provide a lower bounding algorithm based on Lagrangian relaxation and decomposition. Next, we provide "hybrid Lagrangian-LP relaxation" technique which uses the advantages of Lagrangian relaxation and LP relaxation together.

3.3.1.1 Lagrangian Relaxation

In formulation (3.29 - 3.35), only the assignment constraints (3.30) tie the facilities together. The other constraints can be decomposed across facilities, defining a subproblem for each facility i. We relax constraints (3.30) with π_j as Lagrangian multipliers. The objective function of the relaxed problem is the following:

$$\sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i} f_{is}Z_{ins}+\sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i}\sum_{j\in J_i} c_{ij}d_jX_{insj}+\sum_{j\in J}\pi_j\left(1-\sum_{i\in I_j}\sum_{n\in N_i}\sum_{s\in L_i}X_{insj}\right).$$

Combining Lagrangian penalties and transportation costs as $\bar{c}_{ij} = d_jc_{ij} - \pi_j$, we obtain the following as the Lagrangian relaxed problem:

$CCP_{LR}(\pi)$:

$$\min \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i} f_{is}Z_{ins} + \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i}\sum_{j\in J_i}\bar{c}_{ij}X_{insj} + \sum_{j\in J}\pi_j \qquad (3.36)$$

$$\text{subject to } Z_{ins} \geq X_{insj} \quad \forall j \in J, i \in I_j, n \in N_i(\alpha_j), s \in L_i \qquad (3.37)$$

$$\sum_{j\in J} d_jX_{insj} \leq \bar{\mu}_{ins} \quad \forall i \in I, n \in N_i, s \in L_i \qquad (3.38)$$

$$\sum_{n \in N_i} \sum_{s \in L_i} Z_{ins} \leq 1 \quad \forall i \in I \tag{3.39}$$

$$X_{insj} = 0 \text{ or } 1, \quad \forall i \in I, n \in N_i, s \in L_i, j \in J_i \tag{3.40}$$

$$Z_{ins} = 0 \text{ or } 1, \quad \forall i \in I, n \in N_i, s \in L_i. \tag{3.41}$$

In section 3.2.3, we show that constraints (3.39) are redundant for the original problem, but in the relaxed formulation they become useful to improve the bound.

Proposition 7. *Limiting constraints (3.39) improve the Lagrangian relaxation bound.*

Proof. Since the π_j's are unrestricted in sign, the modified assignment costs $\bar{c}_{ij}$ can be negative. A negative transportation cost may compensate the fixed cost of opening a facility for more than a single (n, s) pair, i.e., if $f_{is_1} + \sum_{j \in \hat{J}_{in_1 s_1}} \bar{c}_{ij} \leq 0$ and $f_{is_2} + \sum_{j \in \hat{J}_{in_2 s_2}} \bar{c}_{ij} \leq 0$ for two potential fill rates and stock levels, (n_1, s_1) and (n_2, s_2), in which case the optimal solution can have $Z_{in_1 s_1} = 1$ and $Z_{in_2 s_2} = 1$. Constraints (3.39) eliminate this possibility. □

A solution method for CCP_{LR} is described by the next proposition. First, note that the relaxed problem can be decomposed into facility-level sub-problems P_i, one for each i:

P_i:

$$\min \sum_{n\in N_i}\sum_{s\in L_i} f_{is}Z_{ins} + \sum_{n\in N_i}\sum_{s\in L_i}\sum_{j\in J_i} \bar{c}_{ij}X_{insj} \tag{3.42}$$

$$\text{subject to } Z_{ins} \geq X_{insj} \;\; \forall j \in J_i, n \in N_i(\alpha_j), s \in L_i \tag{3.43}$$

$$\sum_{j\in J_i} d_j X_{insj} \leq \bar{\mu}_{ins} \;\; \forall n \in N_i, s \in L_i \tag{3.44}$$

$$\sum_{n\in N_i}\sum_{s\in L_i} Z_{ins} \leq 1 \tag{3.45}$$

$$X_{insj} = 0 \text{ or } 1, \;\; \forall n \in N_i, s \in L_i, j \in J_i \tag{3.46}$$

$$Z_{ins} = 0 \text{ or } 1, \;\; \forall n \in N_i, s \in L_i. \tag{3.47}$$

With the optimal solutions values $V(P_i)$ to problems P_i, we can compute the optimal value of the Lagrangian relaxation problem as $V(CCP_{LR}(\pi)) = \sum_i V(P_i) + \sum_{j\in J} \pi_j$. The following proposition shows that each facility problem P_i can be further decomposed into base fill rate level and stock level problems, P_{ins}:

Proposition 8. $V(P_i) = \min_{n\in N_i, s\in L_i} \{0, f_{is} + V(P_{ins})\}$ *where*

P_{ins}*:*

$$\min \sum_{j\in J_{in}} \bar{c}_{ij}X_{insj} \tag{3.48}$$

$$\textit{subject to } \sum_{j\in J_{in}} d_j X_{insj} \leq \bar{\mu}_{ins} \tag{3.49}$$

$$X_{insj} = 0 \textit{ or } 1, \;\; \forall j \in J_{in} \tag{3.50}$$

and $J_{in} = \{j \in J_i : \alpha_j \leq b_{in}\}$.

Proof. If we drop constraints (3.45), P_i can be further decomposed into sub-problems P_{ins} for each n and s. Constraints (3.45) state that at most one of those sub-problems can be active. Hence we have:

$$V(P_i) = \min \sum_n \sum_s Z_{ins} \left(f_{is} + V(P_{ins})\right) \tag{3.51}$$

$$\text{subject to} \qquad \sum_n \sum_s Z_{ins} \leq 1 \tag{3.52}$$

Since setting Z_{ins} to 0 is feasible (corresponding to not opening facility (i, n, s)), problem P_i can be solved by solving problems P_{ins} $\forall n \in N_i, s \in L_i$, and setting $V(P_i) = \min_{n \in N_i, s \in L_i} \{0, f_{is} + V(P_{ins})\}$. □

Corollary 3.3.1. *CCP_{LR} is solved by finding solutions to a series of knapsack problems for each facility i, base fill rate level n, and inventory level s as P_{ins} (3.48)-(3.50).*

Using Corollary 3.3.1, we develop Algorithm 1 to solve P_i.

Algorithm 1 Solving P_i

for all $(n, s) \in (N_i, L_{in})$ **do**
    SOLVE P_{ins} for $j \in J_{in}$
    Construct the solution set as $\hat{J}_{ins} = \{j \in J_{in} : X_{insj} = 1\}$
end for
Find (n^*, s^*) such that $V(P_{i,n^*,s^*}) = \min_{n \in N_i, s \in L_{in}} V(P_{ins})$
if $f_{is} + V(P_{i,n^*,s^*}) \leq 0$ **then**
    $V(P_i) = f_{i,s^*} + V(P_{i,n^*,s^*})$
    $Z_{i,n^*,s^*} = 1$ {Open facility i with fill rate b_{i,n^*} and stock level s^*}
    $X_{i,n^*,s^*,j} = 1$ for all $j \in \hat{J}_{in^*s^*}$ {Assign customers to facility i}
else
    Set $V(P_i) = 0$ {Do not open facility i}
end if

In lines 1-4 of Algorithm 1 we solve a knapsack problem for each base fill rate and stock level, and we perform the required updates in lines 5-10 if the facility is open. If the facility is not open (by setting all X_{insj} and Z_{ins} variables for this facility to 0) we set its value to zero in line 11.

The complexity of this algorithm is determined by the number of knapsack problems to be solved and also by the complexity of the knapsack problem itself, which is known to be NP-hard. However, there are practically efficient algorithms such as the dynamic programming algorithm in Kellerer et al. (2005) and Pferschy (1999) for the knapsack problems. One well known algorithm for the knapsack problems is constructed on the idea of dynamically increasing the knapsack capacity, which has a complexity of $O(|J|D_i\log D_i)$, where $D_i = \sum_{j\in J_i} d_j$ is the capacity of the knapsacks (i.e., the total time demand in our case). Note that, overall, we solve $\sum_{n\in N_i} L_{in}$ knapsack problems; thus, keeping L_{in} as tight as possible has significant effect on the solution time. Then, the complexity of solving problem P_i with Algorithm 1 is $O((\sum_{n\in N_i} L_{in})|J|D_i\log D_i)$.

One special case of this problem is constructed when all customers have the same mean demand rate, i.e., $d_j = d, \ \forall j$ (in SPL systems, this may not be too unrealistic, as it represents the case in which each customer has the same number of parts installed). For this setting, the knapsack problem P_{ins} reduces to an easier problem, $P_{ins}(d)$ as:

$$\min \sum_{j\in J_i} \bar{c}_{ij} X_{insj}$$

$$s.t: \sum_{j \in J_{in}} X_{insj} \leq \lfloor \mu_{ins}/d \rfloor$$
$$X_{insj} = 0 \text{ or } 1, \ \ \forall j \in J_{in}$$

where $\lfloor \mu_{ins}/d \rfloor$ is the maximum number of assignments one can make to facility i at fill rate b_{in} and stock level s. This problem is easily solved by ordering the customers with increasing $\bar{c}_{ij}$ values, and then assigning customers to the facility starting from the lowest cost customer until the "knapsack is full" (the maximum number of customers allowed is reached).

The complexity of this algorithm is dominated by the complexity of the sorting customers, which is done in $O(|J|\log|J|)$ time. Then, for this special case, the complexity of Algorithm 1 becomes $O\left(|J|\log|J| + \sum_{n \in N_i} L_{in}\right)$ (for each facility order the customers and then perform assignments for each base fill rate and stock level).

Without the special case, same demand rates, we need to solve many knapsack problems to compute the LR lower bound, i.e., for a problem instance of 100 facilities with 400 customers, on average we solve 15 knapsack problems per facility. Therefore, to control computation time, we focus on decreasing the number and size of knapsack problems to be solved. We employ the following ideas:

i. Note that customers with nonnegative $\bar{c}_{ij}$ can not be in the solution. Hence, we construct set $J_i^- = \{j \in J : \bar{c}_{ij} < 0\}$ (and similarly $J_{in}^- = \{j \in J_i^- : \alpha_j \leq b_{in}\}$) in Algorithm 1 for facility i. This results in smaller knapsack problems (each with smaller number of customers).

ii. Recall that we calculate L_{in} by considering the total lead time demand λ_i of the customers in set J_i (see discussion in Section 3.2.3). Now, with the simplification above, we recalculate the total lead time demand $\lambda_i^- = t_i \sum_{(j \in J_i^-)} d_j$ where $\lambda_i^- \leq \lambda_i$, which leads to a new set of stock levels L_{in}^- with $|L_{in}^-| \leq |L_{in}|$. Hence, the number of knapsack problems to be solved is reduced from $\sum_{n \in N_i} L_{in}$ to $\sum_{n \in N_i} L_{in}^-$.

iii. For the largest stock level for base fill rate n, L_{in}^-, we can simply calculate $V(P_{i,n,L_{in}^-}) = \sum_{j \in J_i^-} \bar{c}_{ij}$, since, for $s = L_{in}^-$, $\sum_{j \in J_i^-} d_j \leq \bar{\mu}_{ins}$ (hence constraints (3.54) is satisfied). Then, we solve knapsack problems P_{ins} up to stock level $L_{in}^- - 1$.

iv. Similarly, we construct the set of base fill rates N_i^- by considering the customers in J_i^- (there may not be any customer in set J_i^- requiring a specific fill rate b_{in}).

In addition to the ideas above, the key idea that will further reduce the number of knapsack problems is *"fathoming"* the knapsack problems by using lower and upper bounds (on knapsack problems). Since we seek the best (n, s) pair for each facility i, by using a good incumbent solution for i (i.e., upper bound of a good $(\hat{n}, \hat{s})$ pair with upper bound $UB(P_{i\hat{n}\hat{s}})$, $UB_i^* = UB(P_{i\hat{n}\hat{s}})$), we can fathom the knapsack problems for all (n, s) with $LB(P_{ins}) \geq UB_i^*$, where $LB()$ and $UB()$ are lower and upper bound values for the corresponding knapsack problems. Note that we can find very tight lower and upper bounds for knapsack problems efficiently. Well known lower bound technique

for the knapsack problem is filling up the capacity starting with the customers having lowest $\bar{c}_{ij}/d_j$ ratio, and ignoring the integrality requirement for the last customer in the knapsack (so we can include some portion of the last customer to fully utilize the capacity). This algorithm gives the same bound as the LP relaxation. Note that if we remove the last customer from the knapsack, we obtain a feasible solution, hence an upper bound. We can further improve this solution by checking the remaining customers to fill the remaining capacity. We apply all these ideas to Algorithm 1 in an efficient manner as shown in the new version, called Algorithm 2.

In line 1 of Algorithm 2, we initialize the incumbent knapsack solution to be 0, since $V(P_i)$ can not be positive (see discussion in Proposition 8), and we also initialize stock and base fill rate levels to be empty. In lines 2-3 we calculate J_i^-, N_i^- , and L_{in}^- by only considering customers having negative costs. We order the customers with respect to their $\bar{c}_{ij}/d_j$ ratios in line 4, and construct ordered sets J_{in}^- for each $n \in N_i^-$ in line 5. Note that while calculating lower bound ($LB(P_{ins})$ in line 17), upper bound ($UB(P_{ins})$ in line 21), and solving for optimality (in line 40) we only consider the customers in set J_{in}^-. In line 6, we initialize the active knapsack problems to be solved for facility i.

As discussed before (in item iii above), in lines 7-15, we calculate $V(P_{ins})$ for the stock level $s = L_{in}^-$ (which can cover demands of all customers in J_{in}^-) for all base fill rates. Then, we update the incumbent by picking up the best (negative) among those. The problems that are solved in this for loop

Algorithm 2 Improved algorithm to solve the problem P_i

$UB_i^* = 0$, $(s^*, n^*) = (\emptyset, \emptyset)$ {Initialize incumbent solution}
$J_i^- = \{j \in J_i : \bar{c}_{ij} < 0\}$ {Set of customers having negative cost}
Calculate N_i^- and L_{in}^-
$J_i^- = \{j \in J_i^- : \bar{c}_{i[k]} \leq \bar{c}_{i[l]} \ \forall k \leq l\}$ {Order the customers with respect to their costs}
$J_{in}^- = \{j \in J_i^- : \alpha_j \leq b_{in}\}$
$A = \{(n, s) : n \in N_i^-, s \in L_{in}^-\}$ {Set of active knapsack problems to be solved}

for all $n \in N_i^-$, $s = L_{in}^-$ **do**
    $V(P_{ins}) = \sum_{j \in J_{in}^-} \bar{c}_{ij}$ {recall that $\mu_{ins} \geq \sum_{j \in J_{in}^-} d_j$ for $s = L_{in}^-$}
    if $f_{is} + V(P_{ins}) < UB_i^*$ **then**
        $UB_i^* = f_{is} + V(P_{ins})$ {update incumbent solution}
        $s^* = s$ and $n^* = n$
        $\hat{J}_{ins} = J_{in}^-$ {record the customers in the solution of (i, n, s)}
    end if
    $A = A \setminus (n, s)$ {FATHOM}
end for

for all $(n, s) \in A$ **do**
    Calculate $LB(P_{ins})$
    if $f_{is} + LB(P_{ins}) \geq UB_i^*$ **then**
        $A = A \setminus (n, s)$ {FATHOM}
    else
        Calculate $UB(P_{ins})$
        if $(UB(P_{ins}) - LB(P_{ins}))/|LB(P_{ins})| < \epsilon$ **then**
            $UB_i^* = f_{is} + UB(P_{ins})$ {update incumbent solution}
            $s^* = s$ and $n^* = n$
            $\hat{J}_{ins} = j \in UB(P_{ins})$ {record the customers in the solution of (i, n, s)}
            $A = A \setminus (n, s)$ {FATHOM}
        else
            if $f_{is} + UB(P_{ins}) < UB_i^*$ **then**
                $UB_i^* = f_{is} + UB(P_{ins})$ {update incumbent solution}
            end if
        end if
    end if
end for

while $A \neq \emptyset$ **do**
    for all $(n, s) \in A$ **do**
        if $f_{is} + LB(P_{ins}) \geq UB_i^*$ **then**
            $A = A \setminus (n, s)$ {FATHOM}
        end if
    end for
    SOLVE $P_{i\hat{n}\hat{s}}$ for $(\hat{n}, \hat{s}) : LB(P_{i\hat{n}\hat{s}}) \leq LB(P_{ins}) \ \forall (n, s) \in A \ j \in J_{in}^-$
    $A = A \setminus (\hat{n}, \hat{s})$ {FATHOM}
    if $f_{is} + V(P_{i\hat{n}\hat{s}}) < UB_i^*$ **then**
        $UB_i^* = f_{is} + V(P_{i\hat{n}\hat{s}})$ {update incumbent solution}
        $s^* = \hat{s}$ and $n^* = \hat{n}$
        $\hat{J}_{i\hat{n}\hat{s}} = j \in V(P_{i\hat{n}\hat{s}})$ {record the customers in the solution of $(i, \hat{n}, \hat{s})$}
    end if
end while

if $(s^*, n^*) \neq (\emptyset, \emptyset)$ **then**
    $V(P_i) = U_i^*$
    $Z_{i,n^*,s^*} = 1$ {Open facility i with fill rate b_{in^*} and stock level s^*}
    $X_{i,n^*,s^*,j} = 1$ for all $j \in \hat{J}_{i,n^*,s^*}$ {Assign customers to facility i}
else
    Set $V(P_i) = 0$ {Do not open facility i}
end if

are then removed from the active set of knapsacks.

In the second for loop (lines 16-33), we calculate a lower bound for each active knapsack, and, if they do not have potential to improve the current incumbent (see the condition in line 18), we fathom these knapsacks, else we calculate their upper bounds, and update the incumbent if needed. If the percentage difference between upper and lower bounds is very small (less than ϵ), then we conclude that the upper bound solution is in fact optimal. For the same fill rate level, while the stock level is increasing, we use the smaller stock level lower bound solution as a base and add new customers for the remaining capacity. In other words, we calculate the lower bound for the knapsack problem of:

$$\min \sum_{j \in \tilde{J}_{in}^{-}} \bar{c}_{ij} X_{insj} \tag{3.53}$$

$$\text{subject to } \sum_{j \in \tilde{J}_{in}^{-}} d_j X_{insj} \leq \bar{\mu}_{ins} - \bar{\mu}_{in,s-1} \tag{3.54}$$

$$X_{insj} = 0 \text{ or } 1, \ \ \forall j \in \tilde{J}_{in}^{-} \tag{3.55}$$

where $\tilde{J}_{in} = \{j \in J_{in}^{-} \setminus \hat{J}_{in,s-1}\}$ and $\hat{J}_{in,s-1}$ are customers in the solution of the knapsack problem with the smaller stock level (note that the last customer in the knapsack problem with stock level $s-1$ can be partially assigned, and hence, $\tilde{J}_{in}$ will have remaining part of this customer).

In lines 35-39, we fathom the knapsacks by using their lower bounds and the current incumbent. If there are still active knapsacks, we pick a candidate problem with the lowest lower bound, and solve it to optimality

by using Xpress-MP solver. Then, we update the incumbent and repeat this procedure until no active knapsack remains. In the final part of the algorithm (lines 48-54), we open facility if it is profitable, i.e., the best knapsack solution value is negative.

Note that, the time spent for lower bound and upper bound calculations is negligible compared to solving a single knapsack problem to optimality, and with the ideas used in Algorithm 2, the number of knapsacks solved is reduced drastically (see results in Section 3.3.3 for details). Still, when we deal with large scale problems, we may need to solve many knapsack problems. In the next section we provide a hybrid scheme, as an alternative to Lagrangian relaxation. In Section 3.3.3 we compare these alternative lower bounding techniques to finalize our methodology development.

3.3.1.2 Hybrid Lagrangian and LP Relaxation

In Section 3.3.1.1, we develop a lower bound method based on Lagrangian relaxation that leads to an efficient algorithm enhanced with fathoming techniques and ideas to decrease the number of knapsack problems. Our purpose in this section is to take advantage of the easily computable LP relaxation bounds of the knapsack subproblems and avoid solving an integer knapsack problem whenever its LP relaxation bound "signals" that the improvement (in the facility-level subproblem's value) due to this knapsack's integer solution is very small. Ultimately, this results in a small loss in the tightness of the lower bounds obtained through Lagrangian relaxation, but

the loss (and the gain in computational time due to unsolved integer knapsack problems) can be controlled through a tolerance parameter which is used to decide whether it is sufficient to use the LP relaxation bound or if solving the integer knapsack problem is needed.

As the decisions about which knapsack problems should be solved to optimality depend on the values of the Lagrangian multipliers and "dynamically" determined in each subgradient iteration, we can view the underlying model as a further relaxation of the Lagrangian relaxed problem, $CCP_{LR}(\pi)$. In this new relaxation, some of the binary variables are treated as continuous (as in the conventional LP relaxation), hence one can view this as a hybrid relaxation between LP relaxation and Lagrangian relaxation, in which the subset of variables to be relaxed is decided in each iteration according to the value of the tolerance parameter and the quality of the solutions. The overall idea is to benefit from both the simplification arising from the LP relaxation (and computational savings) and the tightness of the Lagrangian relaxation. We can formulate the so called hybrid relaxation (HR) as follows:

$CCP_{HR}(\pi)$:

$$\min \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i} f_{is}Z_{ins} + \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i}\sum_{j\in J_i} \bar{c}_{ij}X_{insj} + \sum_{j\in J}\pi_j \tag{3.56}$$

$$\text{subject to } Z_{ins} \geq X_{insj} \quad \forall j \in J, i \in I_j, n \in N_i(\alpha_j), s \in L_i \tag{3.57}$$

$$\sum_{j\in J} d_j X_{insj} \leq \bar{\mu}_{ins} \quad \forall i \in I, n \in N_i, s \in L_i \tag{3.58}$$

$$\sum_{n\in N_i}\sum_{s\in L_i} Z_{ins} \leq 1 \quad \forall i \in I \tag{3.59}$$

$$0 \leq X_{insj} \leq 1, \ \ \forall (i,n,s,j) \in (I, N_i, L_i, J_i)^{LP} \tag{3.60}$$

$$X_{insj} = 0 \text{ or } 1, \ \ \forall (i,n,s,j) \in (I, N_i, L_i, J_i) \setminus (I, N_i, L_i, J_i)^{LP} \tag{3.61}$$

$$Z_{ins} = 0 \text{ or } 1, \ \ \forall i \in I, n \in N_i, s \in L_i. \tag{3.62}$$

This formulation is the same as the formulation of $CCP_{LR}(\pi)$ (3.36)-(3.41), except that constraints (3.60) allow some X_{insj} variables to be non-integer. With this modification, we conclude that

$$V(CCP_{HR}(\pi)) \leq V(CCP_{LR}(\pi)) \leq V(CCP).$$

Note that the subset of the variables that are relaxed to be continuous, X_{insj} $(i,n,s,j) \in (I, N_i, L_i, J_i)^{LP}$ is determined in every iteration (hence dependent on π) and potentially change over the iterations, we cannot compare the π-specific LR and HR bounds with the LP relaxation bound of the overall problem, $V(CCP_{LP})$. If we kept the subset of variables relaxed the same across subgradient iterations, one would have obtained a Lagrangian relaxation dual bound that is tighter than that of LP relaxation. The challenging part of the HR formulation is the definition of set $(I, N_i, L_i, J_i)^{LP}$ in constraints (3.60). If we define this set to be empty, then we have $V(CCP_{LR}(\pi)) = V(CCP_{HR}(\pi))$ for all π and hence even their Lagrangian relaxation duals would be the same. On the other hand, if this set has all possible elements (all X_{insj} variables are non-integer), the HR bound will be closer to the LP bound (if the same variables are kept continuous). We address the challenge of determining this subset by deciding it dynamically (changing it over the iterations), depending on the tolerance parameter and tightness of the bounds. However, even with

changing subsets of variable relaxations, we can prove that the bound obtained through HR is loose as compared to the LR bound, but, as mentioned above, we control the degradation of the lower bound.

Recall that, Algorithm 2 solves knapsack problem P_{ins} to optimality if it has potential to improve the incumbent for facility i, UB_i^* (that is, $f_{is} + LB(P_{ins}) < UB_i^*$). We apply the HR scheme at this point by stating that, if the potential improvement by a knapsack is less than a tolerance, denoted by η, (that is, $UB_i^* - (f_{is} + LB(P_{ins})) < \eta$), then we accept the lower bound solution (which is coming from continuous relaxation of selected X_{insj} variables in this knapsack problem) and use it as the value of problem P_{ins}. Through η, we can control the amount by which the lower bound value disimproves:

Proposition 9. *The worst case performance of HR with respect to LR is*

$$V(CCP_{LR}(\pi)) - V(CCP_{HR}(\pi)) \leq \eta * |I|$$

Proof. If facility i is open with the lower bound solution instead of its optimal solution, then the loss is $V(P_{ins}) - LB(P_{ins})$. We know that $UB_i^* - (f_{is} + LB(P_{ins})) < \eta$ (from the definition of HR), and $f_{is} + V(P_{ins}) \leq UB_i^*$ (from optimality). Hence

$$V(P_{ins}) - LB(P_{ins}) \leq UB_i^* - (f_{is} + LB(P_{ins})) < \eta.$$

Then, if all facilities are open, and if all of their solutions come from the LP relaxations of the knapsack problems, the total loss in the overall lower bound across all facilities is bounded by $\eta * |I|$. □

Note that the worst case loss given in Proposition 9 is very pessimistic, since (1) only a small subset of the facilities are open over all candidates, and (2) the difference $LB(P_{ins}) - V(P_{ins})$ is much smaller than η in general. Hence, HR performs much better in practice. We compare HR with LR in Section 3.3.3.

3.3.2 Upper Bound

In the previous section, we propose methods to find lower bounds for CCP. The next step in our solution methodology is to find an upper bound for the overall problem. Later in Section 3.3.3, we show how we combine lower and upper bound methods in an iterative subgradient algorithm.

At the end of the lower bound algorithm, we obtain a solution for the relaxed problem, which is infeasible since some customers may be assigned to multiple facilities whereas others may not be assigned to any facility at all (note that if the lower bound solution is feasible, then it is optimal). Our purpose in the upper bounding methodology is to use the information obtained from this infeasible lower bound solution and develop a feasible, low-cost solution.

Our upper bound methodology is based on two steps: (1) feasibility restoration, and (2) improvement search.

3.3.2.1 Step 1. Feasibility restoration

In this step we generate a feasible solution, modifying the lower bound solution. We first categorize the customers across the following three groups:

i. $\sum_{i \in I} \sum_{n \in N_i} \sum_{s \in L_i} X_{insj} = 1$, customers assigned to a single facility only.

ii. $\sum_{i \in I} \sum_{n \in N_i} \sum_{s \in L_i} X_{insj} > 1$, customers assigned to multiple facilities.

iii. $\sum_{i \in I} \sum_{n \in N_i} \sum_{s \in L_i} X_{insj} = 0$, customers not assigned to any facility.

The customers in the first group are already feasible for the assignment constraints, hence we keep their assignments as they are in the lower bound solution.

The customers in the second group cause infeasibility because of the multiple assignments. To overcome the infeasibility, for each customer j in this group, we choose the best facility among the facilities that the customer j is assigned, $\hat{I}_j$. We first calculate the cost of assignment $\hat{c}_{ij}$ for the facilities $\hat{I}_j$. We calculate $\hat{c}_{ij}$ by considering the benefit of cancelling the assignment (i, j). Note that $\hat{c}_{ij}$ may have different components:

- If assignment (i, j) is cancelled, we save the transportation cost; hence, $\hat{c}_{ij} += \bar{c}_{ij}$
- According to the other customers that are assigned to i, $\hat{J}_i$, and the capacity usage at facility i, $\sum_{j \in \hat{J}_i} d_j$, cancelling the assignment (i, j) may lead to reducing the stock level and, hence, benefiting from the holding cost. Suppose that the capacity of facility i is $\bar{\mu}_{i\hat{n}\hat{s}}$, where $b_{i\hat{n}} \geq \alpha_j$, $\forall j \in \hat{J}_i$, and $\bar{\mu}_{i\hat{n}\hat{s}} \geq \sum_{j \in \hat{J}_i} d_j$. When the (i, j) assignment is cancelled, we recalculate the required base fill rate level $b_{i\tilde{n}}$ as $\min_{n \in N_i} b_{in} \geq \alpha_j \; \forall j \in \hat{J}_i \setminus$

j (if j is the only customer such that $b_{i\hat{n}} = \alpha_j$, then required base fill rate level reduces), and required stock level $\tilde{s}$ as $\min_{s \in L_{i\tilde{n}}} \bar{\mu}_{i\tilde{n}s} \geq \sum_{j \in \hat{J}_i \setminus j} d_j$. Then, we update $\hat{c}_{ij}$ as $\hat{c}_{ij} += h_i(\hat{s} - \tilde{s})$.

- If j is the only customer that facility i serves, when the assignment (i, j) is cancelled, we also close facility i, hence, $\hat{c}_{ij}$ includes fixed facility opening cost recovered by closing facility i, $\hat{c}_{ij} += f_i$.

Then, we keep the assignment (i^*, j) for customer j, where $\hat{c}_{i^*,j} = \min_{i \in \hat{I}_j} \hat{c}_{ij}$, and cancel the assignments (i, j) for all $i \in \hat{I}_j \setminus i^*$.

The customers in the third group are handled similarly. Since they are not assigned to any facility, we consider all facilities I_j as candidates for these customers, and calculate the cost of assignment $\hat{c}_{ij}$ as follows:

- For candidate facility i, we have the transportation cost, hence $\hat{c}_{ij} += c_{ij}$
- According to the other customers that are assigned to i, $\hat{J}_i$, and the capacity usage at facility i, $\sum_{j \in \hat{J}_i} d_j$, assigning customer j to facility i may require increasing the base fill rate and stock level at the facility. Note that the capacity of facility i is $\bar{\mu}_{i\hat{n}\hat{s}}$, where $b_{i\hat{n}} \geq \alpha_j \ \forall j \in \hat{J}_i$ and $\bar{\mu}_{i\hat{n}\hat{s}} \geq \sum_{j \in \hat{J}_i} d_j$. When assignment (i, j) is included, we recalculate the required base fill rate level $b_{i\tilde{n}}$ as $\min_{n \in N_i} b_{in} \geq \alpha_j \ \forall j \in \hat{J}_i \cup j$, and required stock level $\tilde{s}$ as $\min_{s \in L_{i\tilde{n}}} \bar{\mu}_{i\tilde{n}s} \geq \sum_{j \in \hat{J}_i \cup j} d_j$. Then, we update $\hat{c}_{ij}$ as $\hat{c}_{ij} += h_i(\tilde{s} - \hat{s})$.

- If facility i is not open, hence j is the only customer assigned to it, $\hat{c}_{ij}$ includes fixed facility opening cost too, $\hat{c}_{ij} += f_i$.

Then, we assign each of the customers in this group to the lowest cost facility i^*, where $\hat{c}_{i^*,j} = \min_{i \in I_j} \hat{c}_{ij}$.

3.3.2.2 Step 2. Improvement Search

Note that in the feasibility restoration step, we do not check whether the customers in the first group have lowest cost allocations or not. In addition, we evaluate each customer sequentially, based on the calculations performed on previous customers. Here, we search all possible reallocations of the customers to improve the upper bound solution. We calculate the cost $\hat{c}_{ij}$ of assigning each customer j to facilities in I_j as described in the feasibility restoration step, and then choose the lowest cost facility among them. Note that when we reallocate customer j from one facility to another, say from i_1 to i_2, the total cost will improve as much as $\hat{c}_{i_1 j} - \hat{c}_{i_2 j}$. We repeat this search until no more improvement is possible through customer reassignment.

Note that $\hat{c}_{ij}$ reflects the change in the cost if assignment (i, j) is cancelled (or constructed), but it does not reflect the cost of customer j to the overall system, namely $\sum_{i \in \hat{I}} \sum_{j \in \hat{J}_i} \hat{c}_{ij} \geq \sum_{i \in \hat{I}} f_{i\hat{s}} + \sum_{i \in \hat{I}} \sum_{j \in \hat{J}_i} c_{ij} d_j$. That is, if customers j_1 and j_2 are assigned to facility i, cancelling assignment (i, j_1) will save $\hat{c}_{ij_1}$ and cancelling assignment (i, j_2) will save $\hat{c}_{ij_2}$. However, cancelling both customers j_1 and j_2 will save at most $\hat{c}_{ij_1} + \hat{c}_{ij_2}$. Hence, we name this

type of cost calculation as *"partial cost"* (considering customer j partially, assuming all other customers are fixed in their assignments). This myopic search does not implement a systems.

To have a more "system-oriented" view, we define another cost term, called "*marginal cost*," which is calculated considering the share of each customer in the total system cost. To do this, we calculate the cost of a unit demand at each facility as $p_i = f_{i\hat{s}} / \sum_{j \in \hat{J}_i} d_j$ (note that the facilities with lower "utilization" have higher cost per unit demand). Then, the marginal cost of customer j is calculated as $\check{c}_{ij} = (c_{ij} + p_i)d_j$. Marginal and partial costs have direct opposite influences: We obtain the total system cost by summing the marginal costs $\sum_{i \in \hat{I}} \sum_{j \in \hat{J}_i} \check{c}_{ij} = \sum_{i \in \hat{I}} f_{i\hat{s}} + \sum_{i \in \hat{I}} \sum_{j \in \hat{J}_i} c_{ij} d_j$, but when we move customer j, say from facility i_1 to facility i_2, the difference between marginal costs of customer j at i_1 and at i_2 ($\check{c}_{i_1 j} - \check{c}_{i_2 j}$), does not give the change in the total cost. Therefore, while decreasing the marginal cost of a customer, we may increase the total system cost.

By using marginal costs in our improvement step, we benefit by moving customers from facilities with low utilization (high marginal cost) to facilities with high utilization (low marginal cost). Hence, this approach has tendency to decrease the stock levels and number of facilities. Note that neither cost calculation method is superior to the other, but both have benefits. Therefore, we use both methods, depending on the situation (see Algorithm 3 for more details).

Note that our upper bound methodology handles customers sequen-

tially, one at a time; hence, the evaluation for a customer is based on the calculations and assignments made up to this customer. Therefore, the order in which we evaluate customers affects the solution. Our experiments showed that considering the customers with lower service level requirements (α_j) and high demands (d_j) first has good overall performance.

3.3.3 Subgradient Optimization Algorithm

In this section, we present the overall subgradient-based algorithm (Algorithm 3) which combines lower and upper bound techniques.

In the first part of the Algorithm 3 we have initializations: we set the initial lower bound to 0 in line 1, we calculate the initial upper bound by using several combinations of improvement methods in lines 2-6. Note that we do not have a starting lower bound solution while finding the initial upper bound in line 2; hence, no customer is considered to have an assignment in the upper bound algorithm (see Section 3.3.2, item iii). We refer to the feasibility restoration step as "$find\_feasible(LB)$" where LB stands for the lower bound solution; it returns the feasible upper bound solution. And we refer to the improvement step as "$improve(UB, method)$", where UB stands for the upper bound solution, and $method$ stands for the cost calculation method (note that we use $improve(UB, method1, method2)$ in several steps, which implies we first apply the improvement search with cost calculation $method1$ and then with cost calculation $method2$).

We initialize step size reduction coefficient σ and Lagrangian multipliers

Algorithm 3 Subgradient algorithm

$LB^* = 0$
$UB = find\_feasible(LB^*)$
$UB_1 = improve(UB, partial)$
$UB_2 = improve(UB_1, marginal, partial)$
$UB_3 = improve(UB, marginal, partial)$
$UB^* = min\{UB_1, UB_2, UB_3\}$
Initialize: π, σ
while stopping conditions not satisfied **do**
    Run Algorithm 2 $\forall i \in I$ {returns $V(P_i)$}
    $LB = \sum_{i \in I} V(P_i)$
    if $LB > LB^*$ **then**
        $LB^* = LB$
    end if
    if $LB^+ \leq 2UB^*$ **then**
        $UB = find\_feasible(LB)$
        if $UB \leq 1.5UB^*$ **then**
            if $UB < UB^*$ **then**
                $UB_1 = improve(UB, partial)$
                $UB_2 = improve(UB_1, marginal, partial)$
                $UB_3 = improve(UB, marginal, partial)$
            else
                $UB_1 = improve(UB, marginal, partial)$
            end if
        end if
        $UB^* = min\{UB_1, UB_2, UB_3, UB^*\}$
    end if
    if L^* not improved 20 consecutive iterations **then**
        $\sigma = 0.5\sigma$
    end if
    $\gamma := \sigma(U^* - L) / \sum_j (1 - \sum_i \sum_n \sum_s X_{insj})^2$
    for all $j \in J$ **do**
        $\pi_j := \pi_j + \gamma(1 - \sum_i \sum_n \sum_s X_{insj})$
    end for
end while

π_j in line 7. Initialization of π_j's has an important role in determining a good start in the subgradient search. For the dual problem, π_j represents the profit arising from assigning customer j, and $\sum_{j \in J} \pi_j$ (dual objective) represents the total profit where the right hand sides of the assignment constraints are unit vector. Ignoring the duality gap, we calculate π_j's from the primal-dual optimality condition $\sum_{j \in J} \pi_j = UB^*$, where UB^* is the initial upper bound value obtained in line 6. Hence, π_j corresponds to the cost savings we can have from not assigning customer j. We use the marginal costs of customers calculated in the upper bound solution, and set dual variables (Lagrangian multipliers) as $\pi_j = \check{c}_{\hat{i}j}$ where $\hat{i}$ is the facility to which customer j is assigned in the upper bound solution.

Starting from line 8 until the last step of the Algorithm, we iteratively calculate a lower bound (lines 9-13) and an upper bound (lines 14-26) for a given π, and update parameters σ, γ and π (lines 27-33).

Recall that we have two lower bound methods based on different relaxation techniques (LR and HR), each decomposed into facilities having lower bound cost of LB_i. Later in Section 3.3.4, we compare the performances of the subgradient algorithm with each lower bound technique.

We go into feasibility restoration of the lower bound solution (and calculate upper bounds) in a subgradient iteration only if the actual cost of the current lower bound solution, denoted as LB^+, is less then $2UB^*$ (see Step 14 in Algorithm 3). Here, $LB^+ = \sum_{i \in I} \sum_{n \in N_i} \sum_{s \in L_i} f_{is} \hat{Z}_{ins} + \sum_{i \in I} \sum_{n \in N_i} \sum_{s \in L_i} \sum_{j \in J_i} c_{ij} d_j \hat{X}_{insj}$). The idea here is not to spend time for

upper bound calculations when $LB^+ > 2UB^*$, expecting that the upper bound algorithms will not be able to reduce the cost of the relaxed solution by more than its half. In an iteration in which the feasibility restoration condition is satisfied, we first obtain a feasible solution from the lower bound solution by feasibility restoration. If the feasible upper bound UB is less than 1.5 times best upper bound, then we spend more effort on this upper bound to improve its value further through an improvement search.

We stop the subgradient search when we reach any of the following stopping conditions (listed with how they are represented in Table 3.4):

- When the number of iterations reaches the maximum allowed (i.e., 500 iterations; shown as "*iter*" in Table 3.4).
- When σ is too low (i.e., if $\sigma < 0.001$; shown as "*low* σ" in the table).
- When the gap between best upper and lower bound is small (i.e., if $(UB^* - LB^*)/LB^* < 0.003$; shown as "*gap*" in the table).
- When the run time reaches the limit (i.e., 3600 seconds; shown as "*time*" in the table).

3.3.4 Comparison of Hybrid and Lagrangian Relaxations

In this section, we conduct experiments to compare the lower bound techniques proposed in Sections 3.3.1.1 and 3.3.1.2. We test Algorithm 3 in comparison to Lagrangian relaxation and Hybrid relaxation using the same

data sets described for comparing formulation 1 ($F1$) and formulation 2 ($F2$) in Section 3.2.5. The tolerance value ϵ is fixed to be \$25 in the HR-based lower bound calculation.

We provide lower and upper bound values, the number of subgradient iterations performed, the number of knapsack problems solved, the reason for terminating the algorithm, and the total solution time for both lower bounding techniques in Tables 3.4 and 3.5. In the last two columns of the tables, we calculate the percentage difference between lower bounds $100(LB_{LR} - LB_{HR})/LB_{LR}$ and upper bounds $100(UB_{LR} - UB_{HR})/UB_{LR}$.

With the improvements made in Algorithm 2 (instead of using Algorithm 1), the number of knapsack problems solved in the lower bound calculations is reduced dramatically, i.e., if we were to solve instance a36 using Algorithm 1 as subroutine, more than 5000 knapsack problems need to be solved at a single iteration (hence, the instance needs about 2,500,000 knapsack problems to be solved within 500 iterations). Whereas, with Algorithm 2, we need to solve about 55 knapsack problems per iteration with LR, and 30 knapsack problems per iteration with HR, which means we save about 99% of the effort.

Table 3.4: Comparison of LR and HR with data set a

					LR						HR							
ins	size	d	w (hrs)	h ($)	LB	UB	Iter	Time	Reason	# Knp	LB	UB	Iter	Time	Reason	# Knp	LB Gap	UB Gap
a1	small	low	2	250	30160	30210	72	30	gap	437	30141	30221	80	14	gap	179	0.1	0.0
a2				500	35392	35491	135	65	gap	993	35377	35480	184	40	gap	514	0.0	0.0
a3				1000	45657	45980	500	407	iter	6344	45640	45982	490	179	low σ	2516	0.0	0.0
a4			4	250	28867	28919	183	80	gap	1144	28841	29002	500	30	iter	106	0.1	-0.3
a5				500	32856	32951	311	172	gap	2465	32801	32951	374	29	low σ	217	0.2	0.0
a6				1000	39817	39932	123	90	gap	1348	39839	39906	312	41	gap	393	-0.1	0.1
a7		high	2	250	52740	52775	61	33	gap	520	52747	52775	45	12	gap	165	0.0	0.0
a8				500	61098	61275	88	68	gap	1099	61103	61275	144	73	gap	1131	0.0	0.0
a9				1000	77198	78275	500	549	iter	8838	77167	78275	482	326	low σ	4996	0.0	0.0
a10			4	250	52705	52775	174	109	gap	1576	52678	52775	83	28	gap	369	0.1	0.0
a11				500	60901	61179	500	546	iter	7868	60874	61179	500	314	iter	4153	0.0	0.0
a12				1000	75436	75734	500	835	iter	12341	75329	76102	500	342	low σ	4461	0.1	-0.5
a13	medium	low	2	250	53362	53520	150	347	gap	1735	53339	53499	368	277	gap	1120	0.0	0.0
a14				500	62264	62769	500	2203	iter	11338	62146	62778	500	558	iter	2483	0.2	0.0
a15				1000	79076	81085	500	3072	iter	15764	78892	81795	500	1011	iter	4851	0.2	-0.9
a16			4	250	50524	50693	474	1165	low σ	5412	50483	50798	500	220	iter	539	0.1	-0.2
a17				500	57737	58274	500	2094	iter	9939	57687	58312	500	253	iter	668	0.1	-0.1
a18				1000	71021	72389	500	2772	iter	12807	70885	72389	500	364	iter	1179	0.2	0.0
a19		high	2	250	96085	96226	193	908	gap	4710	95997	96345	500	493	iter	2119	0.1	-0.1
a20				500	110757	111206	500	3065	iter	15818	110648	111711	500	1506	iter	7437	0.1	-0.5
a21				1000	138652	140778	429	3600	time	18743	138371	140673	500	1762	iter	8802	0.2	0.1
a22			4	250	94992	95354	500	2482	iter	11412	94881	95171	500	539	iter	1769	0.1	0.2
a23				500	109273	110051	500	3267	iter	15055	109211	109903	500	1477	iter	6241	0.1	0.1
a24				1000	135675	137562	423	3600	time	16754	135406	137562	500	1635	iter	7056	0.2	0.0
a25	large	low	2	250	81490	83085	187	3600	time	4883	81530	82660	500	1130	iter	579	0.0	0.5
a26				500	94505	98880	160	3600	time	4921	94721	97567	500	1370	iter	885	-0.2	1.3
a27				1000	117599	126452	127	3600	time	5084	119602	124568	500	2541	iter	2480	-1.7	1.5
a28			4	250	80748	85722	139	3600	time	4715	81316	82944	500	1010	iter	317	-0.7	3.2
a29				500	93243	100067	116	3600	time	4762	93987	97482	500	1515	iter	812	-0.8	2.6
a30				1000	113132	123096	116	3600	time	4695	115011	119915	500	2002	iter	1465	-1.7	2.6
a31		high	2	250	148733	151975	113	3600	time	5007	149540	150713	411	3600	time	4201	-0.5	0.8
a32				500	175666	181803	103	3600	time	4998	175890	180917	213	3600	time	4707	-0.1	0.5
a33				1000	223672	235713	101	3600	time	4944	224752	234464	167	3600	time	4754	-0.5	0.5
a34			4	250	148517	152729	96	3600	time	4274	149484	151046	321	3600	time	3289	-0.7	1.1
a35				500	171301	187401	83	3600	time	4411	175897	181081	195	3600	time	3844	-2.7	3.4
a36				1000	216875	236969	80	3600	time	4422	223793	236319	134	3600	time	4168	-3.2	0.3

Table 3.5: Comparison of LR and HR with data set b

					LR						HR							
ins	size	d	w (hrs)	h ($)	LB	UB	Iter	Time	Reason	# Knp	LB	UB	Iter	Time	Reason	# Knp	LB Gap	UB Gap
b1	small	low	2	250	31914	32006	236	131	gap	2012	31863	32006	500	135	iter	1906	0.2	0.0
b2				500	37371	37760	500	288	iter	4987	37360	37871	500	198	iter	2904	0.0	-0.3
b3				1000	47964	49016	500	431	iter	6671	47894	49016	394	185	low σ	2790	0.1	0.0
b4			4	250	28835	28908	130	48	gap	785	28824	28908	139	13	gap	123	0.0	0.0
b5				500	32619	32658	112	57	gap	839	32572	32656	149	23	gap	278	0.1	0.0
b6				1000	39598	39686	194	90	gap	1485	39568	39686	99	26	gap	336	0.1	0.0
b7		high	2	250	54467	54629	61	36	gap	570	54476	54629	215	61	gap	868	0.0	0.0
b8				500	63226	63415	185	216	gap	3935	63215	63415	500	288	iter	4494	0.0	0.0
b9				1000	79750	81275	457	757	low σ	12190	79597	80889	500	414	iter	6481	0.2	0.5
b10			4	250	53341	53520	500	373	iter	5476	53313	53520	500	198	iter	2676	0.1	0.0
b11				500	61324	61484	242	229	gap	3366	61316	61467	331	153	gap	2073	0.0	0.0
b12				1000	75751	77347	500	711	iter	10120	75629	77161	500	232	iter	3148	0.2	0.2
b13	medium	low	2	250	51336	51670	500	1505	iter	7590	51265	51650	500	128	iter	193	0.1	0.0
b14				500	59618	60433	500	2384	iter	12038	59470	60424	500	191	iter	526	0.2	0.0
b15				1000	75576	77374	500	2837	iter	15205	75405	77073	500	446	iter	1840	0.2	0.4
b16			4	250	48953	49068	228	517	gap	2417	48922	49079	500	286	iter	969	0.1	0.0
b17				500	56173	56400	500	1661	iter	8204	56131	56476	500	385	iter	1430	0.1	-0.1
b18				1000	69867	71614	500	2307	iter	10738	69806	71808	500	781	iter	3197	0.1	-0.3
b19		high	2	250	90764	91032	128	526	gap	2704	90673	90944	240	260	gap	1158	0.1	0.1
b20				500	105396	105577	474	2837	gap	14735	105240	105961	500	1015	iter	4888	0.1	-0.4
b21				1000	133336	135277	457	3600	time	18635	133170	135032	500	1991	iter	10009	0.1	0.2
b22			4	250	90769	91038	217	1054	gap	4854	90670	90926	226	362	gap	1392	0.1	0.1
b23				500	105294	105565	464	3128	gap	14596	105151	105584	500	1248	iter	5290	0.1	0.0
b24				1000	130250	133406	444	3600	time	17258	130081	132658	500	1458	iter	6098	0.1	0.6
b25	large	low	2	250	81303	82894	190	3600	time	4885	81370	82385	500	848	iter	150	-0.1	0.6
b26				500	94270	97852	157	3600	time	5042	94439	96619	500	1207	iter	750	-0.2	1.3
b27				1000	117819	127970	122	3600	time	5078	119069	124797	500	3431	iter	3953	-1.1	2.5
b28			4	250	78626	90799	133	3600	time	4874	81179	82236	500	998	iter	150	-3.2	9.4
b29				500	93363	97831	126	3600	time	4779	93876	96443	500	1296	iter	616	-0.5	1.4
b30				1000	116141	127582	112	3600	time	4751	117558	123148	500	2360	iter	2007	-1.2	3.5
b31		high	2	250	147318	156129	112	3600	time	5118	150464	151501	500	3600	time	3854	-2.1	3.0
b32				500	172082	185253	97	3600	time	5144	175865	177889	405	3600	time	4125	-2.2	4.0
b33				1000	222093	230676	101	3600	time	4986	222439	230194	179	3600	time	4753	-0.2	0.2
b34			4	250	150241	152534	105	3600	time	4316	150431	152062	371	3600	time	3378	-0.1	0.3
b35				500	168030	192676	92	3600	time	4523	175641	179023	220	3600	time	3899	-4.5	7.1
b36				1000	218334	236130	86	3600	time	4438	222206	231705	140	3600	time	4209	-1.8	1.9

Note that the LR based lower bound algorithm still requires too many knapsack problems to be solved, resulting 28 instances not completing 500 iterations before reaching the run time limit (hence stopping criteria is time for these instances, e.g., a24-a36). However, HR solves even fewer knapsack problems, so that only 12 instances terminate because of the time limitation (e.g., a31-a36).

Resulting from the reductions in the number of knapsack problems solved, the HR-based lower bound algorithm has an average time per iteration of about 3 seconds, as compared with 7.5 seconds for the LR-based algorithm. Hence we can conclude that the HR-based algorithm is 2.5 times faster than the LR-based algorithm, which especially shows the benefits with large instances. Note that, in all of the large instances (a25-a36, b25-b36), HR has both better lower bounds (up to 3.2%) and also better upper bounds (up to 7.1%). Even for the small and medium instances HR performs as good as LR; and, in the worst case (see the instance a15), the HR upper bound is 0.9% worse than the LR upper bound. But note that LR spends 3072 seconds, while HR spends only 1011 seconds on this instance.

Overall, HR gives either better bounds in the same or shorter time, or it gives bounds very close to those of LR within much shorter times. Hence, we conclude that HR is superior to LR.

3.3.4.1 Results from comparison of F1 and F2 vs HR

In Tables 3.6 and 3.7, we pick the best solution between formulation 1 and formulation 2 (so called best of formulations as BF), and compare it with the HR solution for each instance (recall that, neither of the formulations, F1 or F2, is superior to the other, see discussion in Section 3.2.5). In these tables, we provide best upper bound solutions, their solution times, the number of open facilities $|\hat{I}|$, the total stock levels $|\hat{s}|$ and the upper bound gaps as $(100(UB_{HR} - UB_{BF})/UB_{BF})$.

For small instances, the upper bounds given by BF and HR are very close and, in almost all of these instances (other than b3 and b12), the HR upper bound is worse than the BF upper bound with a gap always less than 1%.

For medium size instances, there is no winner in terms of upper bounds; HR has better upper bounds for some instances (up to 2.5% a18) and BF has better upper bounds for some others (up to 1.3%). However, in terms of solution times, HR performs much better, and, on average, HR spends 777 seconds and BF spends 2450 seconds per medium size instance.

For the large instances, HR gives better upper bounds in almost all instances (other than a31 with a gap of 0.1%, and b25 with a gap of 0.6%), and improves the upper bound of BF up to 41.2%. This shows that HR excels in large scale problems with additional savings in computation times.

Table 3.6: Comparison of best of $F1$ and $F2$ vs HR with data a

					F1 and F2					HR				
ins	size	d	w (hrs)	h ($)	Best UB	Time	Status	$\|\hat{I}\|$	$\|\hat{s}\|$	UB	Time	$\|I\|$	$\|s\|$	UB Gap
a1	small	low	2	250	30210	1	Opt	11	21	30221	14	11	21	0.0
a2				500	35460	2	Opt	11	21	35480	40	11	21	0.1
a3				1000	45960	4	Opt	11	21	45982	179	11	21	0.0
a4			4	250	28919	18	Opt	6	16	29002	30	6	16	0.3
a5				500	32889	53	Opt	6	15	32951	29	6	15	0.2
a6				1000	39906	57	Opt	5	14	39906	41	5	14	0.0
a7		high	2	250	52775	1	Opt	11	34	52775	12	11	34	0.0
a8				500	61275	6	Opt	11	34	61275	73	11	34	0.0
a9				1000	78016	163	Opt	11	33	78275	326	11	34	0.3
a10			4	250	52775	11	Opt	11	34	52775	28	11	34	0.0
a11				500	61079	604	Opt	11	32	61179	314	10	33	0.2
a12				1000	75628	2769	Opt	9	28	76102	342	9	27	0.6
a13	medium	low	2	250	53499	36	Opt	16	37	53499	277	16	37	0.0
a14				500	62749	1004	Opt	16	37	62778	558	16	37	0.0
a15				1000	80723	3644	N/F	16	34	81795	1011	16	37	1.3
a16			4	250	50602	1252	Opt	11	31	50798	220	11	31	0.4
a17				500	57939	3620	N/F	10	28	58312	253	10	30	0.6
a18				1000	74209	3615	N/F	10	27	72389	364	9	27	-2.5
a19		high	2	250	96196	45	Opt	19	61	96345	493	19	61	0.2
a20				500	111332	3600	N/F	18	58	111711	1506	18	59	0.3
a21				1000	142984	3600	N/F	19	55	140673	1762	17	57	-1.6
a22			4	250	95116	968	Opt	18	61	95171	539	17	60	0.1
a23				500	111791	3600	N/F	17	59	109903	1477	14	56	-1.7
a24				1000	137158	3600	N/F	13	52	137562	1635	13	54	0.3
a25	large	low	2	250	82899	3600	N/F	21	59	82660	1130	19	58	-0.3
a26				500	118778	3600	N/F	29	78	97567	1370	18	55	-17.9
a27				1000	154298	3600	N/F	26	78	124568	2541	16	54	-19.3
$a28^1$			4	250	120452	3600	N/F	35	63	82944	1010	19	58	-31.1
$a29^1$				500	149273	3600	N/F	42	77	97482	1515	18	57	-34.7
$a30^1$				1000	203801	3600	N/F	52	86	119915	2002	12	46	-41.2
a31		high	2	250	150592	3600	N/F	31	115	150713	3600	32	118	0.1
a32				500	187268	3600	N/F	33	114	180917	3600	29	113	-3.4
$a33^1$				1000	283862	3600	N/F	43	105	234464	3600	22	102	-17.4
$a34^1$			4	250	195209	3600	N/F	68	162	151046	3600	30	116	-22.6
$a35^1$				500	234224	3600	N/F	64	158	181081	3600	28	113	-22.7
$a36^1$				1000	346458	3600	N/F	62	135	236319	3600	24	100	-31.8

Table 3.7: Comparison of best of $F1$ and $F2$ vs HR with data b

					F1 and F2					HR				
ins	size	d	w (hrs)	h ($)	Best UB	Time	Status	$\|\hat{I}\|$	$\|\hat{s}\|$	UB	Time	$\|I\|$	$\|s\|$	UB Gap
b1	small	low	2	250	31983	3	Opt	11	23	32006	135	11	23	0.1
b2				500	37521	2	Opt	11	22	37871	198	11	23	0.9
b3				1000	48521	4	Opt	11	22	49016	185	12	22	1.0
b4			4	250	28881	8	Opt	7	17	28908	13	6	15	0.1
b5				500	32656	9	Opt	6	15	32656	23	6	15	0.0
b6				1000	39686	25	Opt	5	14	39686	26	5	14	0.0
b7		high	2	250	54629	2	Opt	13	37	54629	61	13	37	0.0
b8				500	63415	15	Opt	12	35	63415	288	12	35	0.0
b9				1000	80467	238	Opt	13	33	80889	414	12	34	0.5
b10			4	250	53411	27	Opt	11	33	53520	198	11	34	0.2
b11				500	61369	281	Opt	11	31	61467	153	11	31	0.2
b12				1000	76269	3600	N/F	9	27	77161	232	9	29	1.2
b13	medium	low	2	250	51519	1894	Opt	14	34	51650	128	14	35	0.3
b14				500	59807	1367	Opt	14	33	60424	191	14	35	1.0
b15				1000	76830	3600	N/F	14	32	77073	446	14	33	0.3
b16			4	250	48994	360	Opt	11	30	49079	286	11	30	0.2
b17				500	56520	3600	N/F	11	30	56476	385	10	29	-0.1
b18				1000	73142	3600	N/F	11	28	71808	781	8	26	-1.8
b19		high	2	250	90890	190	Opt	18	60	90944	260	18	61	0.1
b20				500	105503	2394	Opt	17	57	105961	1015	17	58	0.4
b21				1000	136857	3600	N/F	19	56	135032	1991	17	57	-1.3
b22			4	250	90884	2409	Opt	16	59	90926	362	17	60	0.0
b23				500	105705	3600	N/F	16	57	105584	1248	16	58	-0.1
b24				1000	134057	3600	N/F	16	52	132658	1458	12	50	-1.0
b25	large	low	2	250	81872	3600	N/F	18	55	82385	848	18	58	0.6
$b26^1$				500	130165	3600	N/F	39	70	96619	1207	18	57	-25.8
$b27^1$				1000	166456	3600	N/F	39	67	124797	3431	17	55	-25.0
$b28^1$			4	250	118857	3600	N/F	35	76	82236	998	17	57	-30.8
$b29^1$				500	154471	3600	N/F	46	69	96443	1296	17	55	-37.6
$b30^1$				1000	206096	3600	N/F	47	67	123148	2360	13	51	-40.2
b31		high	2	250	154180	3600	N/F	33	113	151501	3600	29	111	-1.7
b32				500	181316	3600	N/F	29	102	177889	3600	24	102	-1.9
$b33^1$				1000	282311	3600	N/F	46	105	230194	3600	22	99	-18.5
$b34^1$			4	250	208510	3600	N/F	80	180	152062	3600	27	109	-27.1
$b35^1$				500	251581	3600	N/F	59	117	179023	3600	24	102	-28.8
$b36^1$				1000	354921	3600	N/F	55	108	231705	3600	19	97	-34.7

3.4 Conclusion

In this chapter, we study the service parts logistics integrated network design and inventory problem with customer-centric time-based service levels. We consider inventory decisions (stock levels) and their costs explicitly in a network design model, thus making service levels (part availabilities or fill rates) vary across facilities and parts to achieve each customer's time-based service level requirement.

We first provide different MIP formulations, applicable to small and medium size problems. We compare these formulations and highlight the advantages of each. For large problems, we develop a Lagrangian relaxation based algorithm, enhanced with improvement techniques. We further improve the Lagrangian relaxation based algorithm, by applying the so-called "hybrid relaxation," which uses the LP lower bounds of the knapsack sub-problems when possible to attack truly large scale problems. Our results show the effectiveness of the overall approach producing provably near optimal solutions with relatively short computation times.

Chapter 4

System-Wide Service Levels Revisited

4.1 Introduction

In this chapter, we revisit our original system-wide service level problem. In Chapter 2, we show that the integrated approach has significant benefits compared to the traditional decoupled approach. However, the proposed integrated model takes too much time to solve (more than 10 hours in some instances), and the largest size instance we solved in our experiments has about 20 facilities and 150 customers with a very low demand (the mean annual demand rate was about 0.5 units).

In Chapter 3, we focused on the customer-centric service level problem, a special case of the integrated time-based service level problem, for which we propose efficient algorithms capable of providing tight lower and upper bounds for large instances (100 facilities and 400 customers) with higher demands in less than 1 hour.

The customer-centric service level problem has real life applications, especially in after market service parts logistics. However, our ultimate goal in this research is to provide a solution methodology for the system-wide service level problem which is capable of solving large instances in a reasonable amount

of time. In this chapter, we apply the successful methodology used in the customer-centric service level problem in Chapter 3 to the original system-wide service level problem.

We make the following additional assumptions to those made in Chapter 2 that facilitate model development:

- We assume that it is not possible to split a part's demand at a demand point while allocating it to facilities; for example, demand should be satisfied by a single source. Due to the problem's structure, multiple sourcing does not happen very often in optimal solutions.
- We assume that there is a single part in the overall system.
- We finally assume that a customer's demand can be assigned to a facility only if that facility keeps a positive stock level.

4.2 Mathematical Formulation

We first describe the approaches we used for solving the system-wide service level problem by comparing it with the customer-centric service level problem in Section 4.2.1. We then propose a new formulation based on the customer centric model in Section 4.2.2.

4.2.1 Comparing the System-wide and Customer-centric Problems

With the modifications based on the assumptions given in Section 4.1, the system-wide service level problem for a single part, denoted by OM, is as

follows:

Original Model ***(OM)***:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} d_j X_{ij} + \sum_{i \in I} h_i S_i \tag{4.1}$$

$$\sum_{i \in I} X_{ij} = 1, \ \forall j \tag{4.2}$$

$$X_{ij} \leq Y_i, \ \forall i, j \tag{4.3}$$

$$S_i \leq L Y_i, \ \forall i \tag{4.4}$$

$$\sum_{i \in I} \sum_{j \in J} \frac{\delta_{ij} d_j X_{ij}}{d} \beta_i(S_i, \lambda_i) \geq \alpha \tag{4.5}$$

$$\lambda_i = t_i \sum_{j \in J} d_j X_{ij}, \ \forall i \tag{4.6}$$

$$\beta_i(S_i, \lambda_i) = \sum_{r=0}^{S_i - 1} \lambda_i^r \frac{e^{-\lambda_i}}{r!}, \ \forall i \tag{4.7}$$

$$X_{ij} = 0 \text{ or } 1, \ \forall i \in I, j \in J \tag{4.8}$$

$$Y_i = 0 \text{ or } 1, \ \forall i \in I \tag{4.9}$$

$$S_i \in \{0, 1, 2, \ldots, L\}, \ \forall i. \tag{4.10}$$

Here, we use binary assignment variables as in constraints (4.8), as opposed to the continuous assignment variables used in Chapter 2. Note that this formulation (OM) is again nonlinear due to the service level constraints (4.5) and the fill rate constraints (4.7). Therefore, even for this single part version, we could not provide any direct solution technique.

Employing an approach similar to the one used in Chapter 2, we can linearize OM by using the step-wise approximation of the fill rate function as follows:

Linearized Model ***(LM)***:

$$\min \sum_{i\in I} f_i Y_i + \sum_{i\in I}\sum_{j\in J} c_{ij} d_j X_{ij} + \sum_{i\in I}\sum_{s\in L} sh_i W_{is} \qquad (4.11)$$

$$\sum_{i\in I} X_{ij} = 1,\ \forall j \qquad (4.12)$$

$$X_{ij} \le \sum_{s\in L} W_{is},\ \forall i,j \qquad (4.13)$$

$$\sum_{s\in L} W_{is} \le Y_i,\ \forall i \qquad (4.14)$$

$$\lambda_i = t_i \sum_{j\in J} d_j X_{ij},\ \forall i \qquad (4.15)$$

$$R_{i0} = 0,\ \forall i \qquad (4.16)$$

$$a_n - \lambda_i \ge M_1(R_{in} - 1),\ \forall i,n \qquad (4.17)$$

$$a_n - \lambda_i \le M_2 R_{in},\ \forall i,n \qquad (4.18)$$

$$Q_{in} = R_{i,n} - R_{i,n-1},\ \forall i,n \qquad (4.19)$$

$$V_{isn} \le M_3 Q_{in},\ \forall i,s,n \qquad (4.20)$$

$$V_{isn} \le M_3 W_{is},\ \forall i,s,n \qquad (4.21)$$

$$V_{isn} \ge 0\ \forall i,s,n \qquad (4.22)$$

$$V_{isn} \le \sum_{j\in J} \frac{\delta_{ij} d_j X_{ij}}{d} b_{sn},\ \forall i,s,n \qquad (4.23)$$

$$V_{isn} \ge \sum_{j\in J} \frac{\delta_{ij} d_j X_{ij}}{d} b_{sn} - M_3(1 - Q_{in}) - M_3(1 - W_{is}),\ \forall i,s,n \qquad (4.24)$$

$$\sum_{i\in I}\sum_{s\in L}\sum_{n\in N} V_{isn} \ge \alpha \qquad (4.25)$$

$$Y_i, X_{ij}, W_{iks}, Q_{in}, R_{in} = 0 \text{ or } 1,\ \forall i,j,s,n. \qquad (4.26)$$

By using a conservative (underestimated) fill rate approximation (denoted by $\underline{\beta}$ instead of true fill rate function), an optimal solution value of LM

provides an upper bound on the optimal solution of the original model, i.e., $V(OM) \leq V(LM(\underline{\beta}))$. One can think of the underestimated approximation of fill rate as drawing the step-wise function just below the true fill rate curve. Simiarly, by using an overestimated fill rate approximation (say $\bar{\beta}$), LM's solution provides a lower bound for the original model, $V(LM(\bar{\beta})) \leq V(OM)$. Therefore, the interval between the lower and upper bounds obtained through fill rate approximations provides a gap within which the optimal value $V(OM)$ must lie. In fact, by increasing the granularity of the step-wise approximation (i.e., increasing the number of steps and shorten the demand and fill rate intervals), we can make the two approximations $\underline{\beta}$ and $\bar{\beta}$ closer to each other, and tighten the gap between $V(LM(\underline{\beta}))$ and $V(LM(\bar{\beta}))$.

By using the linearized model LM in Chapter 2, we show that we can have significant cost savings by considering inventory decisions together with network design decisions. However, LM is not sufficient to solve large instances, and it may even take hours to find optimal solutions for medium size instances. In Chapter 2, we show that there is ample room for improvement with respect to the solution methods for the integrated approach to the system-wide service level problem, and concluded that further research in this area would be worth pursuing.

In Chapter 3, we focus on a special case of the problem, the customer-centric service level problem. This problem requires a certain time-based service level to be satisfied for each customer (instead of a weighted average service level in the overall system as in the system-wide problem). Our start-

ing point for the customer-centric problem is a nonlinear model based on OM as follows:

$$\min \sum_{i \in I} f_i Y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} d_j X_{ij} + \sum_{i \in I} h_i S_i \tag{4.27}$$

$$\text{subject to} \quad \sum_{i \in I} X_{ij} = 1, \ \forall j \tag{4.28}$$

$$X_{ij} \leq Y_i \ \forall i, j \tag{4.29}$$

$$S_i \leq L Y_i, \ \forall i \tag{4.30}$$

$$\lambda_i = t_i \sum_{j \in J} d_j X_{ij}, \ \forall i \tag{4.31}$$

$$\beta_i = \sum_{r=0}^{S_i - 1} \lambda_i^r \frac{e^{-\lambda_i}}{r!}, \ \forall i \tag{4.32}$$

$$\sum_{i \in I} \beta_i \delta_{ij} X_{ij} \geq \alpha_j, \ \forall j \tag{4.33}$$

$$X_{ij} = 0 \text{ or } 1, \ \forall i, j \tag{4.34}$$

$$Y_i = 0 \text{ or } 1, \ \forall i \tag{4.35}$$

$$S_i \in \{0, 1, 2, \ldots, L\}, \ \forall i. \tag{4.36}$$

This formulation's main difference in comparison to OM is its having a service level constraint (4.33) for each customer. This model is still nonlinear and hence cannot be solved directly. For constraints (4.33) to satisfy, each customer has to be served by a facility having a fill rate higher than the customer's target service level ($\beta_i \geq \max_{j \in \hat{J}_i} \{\alpha_j\}$, where $\hat{J}_i$ are customers served by facility i). Using this idea, we propose a linear mixed integer programming model for the customer-centric service level problem as follows:

$$\min \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i} f_{is}Z_{ins} + \sum_{i\in I}\sum_{n\in N_i}\sum_{s\in L_i}\sum_{j\in J_i} c_{ij}d_jX_{insj} \tag{4.37}$$

$$\text{subject to} \quad \sum_{i\in I_j}\sum_{n\in N_i(\alpha_j)}\sum_{s\in L_i} X_{insj} = 1, \ \ \forall j \in J \tag{4.38}$$

$$Z_{ins} \geq X_{insj}, \ \ \forall j \in J, i \in I_j, n \in N_i(\alpha_j), s \in L_i \tag{4.39}$$

$$t_i \sum_{j\in J_i} d_jX_{insj} \leq \mu_{ins}, \ \ \forall i \in I, n \in N_i, s \in L_i \tag{4.40}$$

$$\sum_{n\in N_i}\sum_{s\in L_i} Z_{ins} \leq 1, \ \ \forall i \in I \tag{4.41}$$

$$X_{insj} = 0 \text{ or } 1, \ \ \forall i \in I, n \in N_i, s \in L_i, j \in J_i. \tag{4.42}$$

One advantage of this model is its inherent flexibility for decomposition. In Chapter 3, we provide tight lower and upper bounds for large instances by using an algorithm based on the Lagrangian relaxation of this model.

4.2.2 New Formulation

In this section, we provide a new formulation for the system-wide service level problem based on the customer-centric formulation (4.37)-(4.42). Hence, we call this new formulation the customer-centric based model, CM.

In a customer-centric service level problem, each customer has to be served by a facility in the time window coverage area, and fill rates at the facilities are forced by individual customers' target service levels α_j as in constraints (4.39). However, in the system-wide service level problem a customer may be assigned to any facility (even to facilities out of the time window coverage), and the facilities' fill rates are determined independently from individual

customers, by choosing from the potential fill rates set N. In other words, the driving force for the facility fill rates are not the individual customers, but the target of the overall system. The model in a way optimally allocates the system-wide target service level to individual customers so that the total cost is minimized.

With similar assignment variables X_{insj}, the achieved time-based service level in the overall system can be calculated as

$$\sum_{i\in I}\sum_{n\in N}\sum_{s\in L}\sum_{j\in J}(\delta_{ij}d_j b_n)X_{insj},$$

which is the covered customer demand $\delta_{ij}d_j$ multiplied by the fill rate of the customer's facility b_n summed across all demands and facilities. Recall that b_n's are predetermined values calculated by step-wise approximation of the fill rate, and L is the maximum potential stock level calculated by considering overall system demand as discussed in Chapter 2. Hence, we can construct the system-wide service level constraint by stating that the achieved service must be greater than the target level, $\alpha\sum_{j\in J}d_j$. With these modifications, we can formulate the system-wide problem based on the customer-centric development. The new formulation is denoted by CM and listed as follows:

$$\min \sum_{i\in I}\sum_{n\in N}\sum_{s\in L} f_{is}Z_{ins} + \sum_{i\in I}\sum_{n\in N}\sum_{s\in L}\sum_{j\in J} c_{ij}d_j X_{insj} \tag{4.43}$$

$$\text{subject to} \quad \sum_{i\in I}\sum_{n\in N}\sum_{s\in L} X_{insj} = 1 \ \ \forall j \in J \tag{4.44}$$

$$Z_{ins} \geq X_{insj} \ \ \forall j \in J, i \in I, n \in N, s \in L \tag{4.45}$$

$$\sum_{j\in J} d_j X_{insj} \leq \bar{\mu}_{ins} \;\; \forall i \in I, n \in N_i, s \in L \tag{4.46}$$

$$\sum_{n\in N}\sum_{s\in L} Z_{ins} \leq 1 \;\; \forall i \in I \tag{4.47}$$

$$\sum_{i\in I}\sum_{n\in N}\sum_{s\in L}\sum_{j\in J}(\delta_{ij} d_j b_n) X_{insj} \geq \alpha \sum_{j\in J} d_j \tag{4.48}$$

$$X_{insj} = 0 \text{ or } 1, \;\; \forall i \in I, n \in N, s \in L, j \in J \tag{4.49}$$

$$Z_{ins} = 0 \text{ or } 1, \;\; \forall i \in I, n \in N, s \in L. \tag{4.50}$$

Here, the first term in the objective (4.43) is the fixed facility cost including the inventory stocking cost with costs $f_{is} = f_i + sh_i$. Z_{ins} is 1 if facility i is open with stock level s and base fill rate b_n. The second term is the transportation cost for assignments X_{insj}. Constraints (4.44) assign all the demands to a facility while constraints (4.45) mean that any facility serving a demand point should be open. Constraints (4.46) are capacity constraints, where $\bar{\mu}_{ins} = \mu_{ins}/t_i$ is the maximum demand that facility i can serve with stock level s while guaranteeing a base fill rate of b_n. Constraints (4.47) limit any facility not open more than once, which is redundant when all potential fill rates are greater than 0.5 (see Proposition 5 for discussion on redundancy of these constraints). Constraint (4.48) is the system-wide service level constraint. Note that we do not have a facility dependent set of stock levels L_i or a facility dependent set of base fill rates N_i in *CM*, because each can be assigned to any facility and a facility may have a fill rate high or low depending on its contribution to the system-wide service level. Hence, we have a maximum stock level L and maximum base fill rate N for the overall system. This requirement for having a more general fill rate and stock level sets in

the CM model results in more variables and constraints as compared to the customer-centric model.

When the set of base fill rates in CM and the fill rate break points in the stepwise approximation in LM are same, then $V(CM) = V(LM)$. Therefore, by picking up a finite set of over and under estimated base fill rates $\bar{\beta}$ and $\underline{\beta}$, CM also provides lower and upper bounds for the OM, i.e., $V(CM(\bar{\beta})) \leq V(OM) \leq V(CM(\underline{\beta}))$. In the following section, we develop a solution methodology based on the decomposition of CM.

4.3 Methodology

In Section 4.3.1, we propose a lower bounding technique based on the Lagrangian relaxation and decomposition of CM. Note that $V(CM_{LR}(\bar{\beta})) \leq V(CM(\bar{\beta}))$ and $V(CM(\bar{\beta})) \leq V(OM)$, hence $V(CM_{LR}(\bar{\beta})) \leq V(OM)$. Then, in Section 4.3.2, we develop an upper bound algorithm using a revised version of the decoupled approach (introduced in Chapter 2) with a feedback mechanism. By combining these techniques with a subgradient algorithm we provide tight gaps for the optimum solution of OM, which makes the overall approach applicable for large instances.

4.3.1 Lower Bound

In formulation (4.43 - 4.50), assignment constraints (4.44) and also the service level constraint (4.48) tie the facilities together. The other constraints can be decomposed across facilities, defining a subproblem for each facility i.

We relax constraints (4.44) with π_j and constraint (4.48) with η as Lagrangian multipliers. The objective function of the relaxed problem is the following:

$$\sum_{i\in I}\sum_{n\in N}\sum_{s\in L} f_{is}Z_{ins} + \sum_{i\in I}\sum_{n\in N}\sum_{s\in L}\sum_{j\in J} c_{ij}d_jX_{insj}$$

$$+\sum_{j\in J}\pi_j\left(1-\sum_{i\in I}\sum_{n\in N}\sum_{s\in L}X_{insj}\right)$$

$$+\eta\left(G-\sum_{i\in I}\sum_{n\in N}\sum_{s\in L}\sum_{j\in J}(\delta_{ij}d_jb_n)X_{insj}\right)$$

where $G=\alpha\sum_{j\in J}d_j$.

Combining Lagrangian multipliers and transportation costs in $\bar{c}_{ijn} = d_jc_{ij}-\pi_j-\eta\delta_{ij}d_jb_n$, we obtain the following as the Lagrangian relaxed problem:

$CM_{LR}(\pi,\eta)$:

$$\min \sum_{i\in I}\sum_{n\in N}\sum_{s\in L} f_{is}Z_{ins} + \sum_{i\in I}\sum_{n\in N}\sum_{s\in L}\sum_{j\in J}\bar{c}_{ijn}X_{insj} + \sum_{j\in J}\pi_j + \eta G \quad (4.51)$$

$$\text{subject to} \quad Z_{ins} \geq X_{insj} \;\; \forall j\in J, i\in I, n\in N, s\in L \quad (4.52)$$

$$\sum_{j\in J} d_jX_{insj} \leq \bar{\mu}_{ins} \;\; \forall i\in I, n\in N_i, s\in L \quad (4.53)$$

$$\sum_{n\in N}\sum_{s\in L} Z_{ins} \leq 1 \;\; \forall i\in I \quad (4.54)$$

$$X_{insj} = 0 \text{ or } 1, \;\; \forall i\in I, n\in N, s\in L, j\in J \quad (4.55)$$

$$Z_{ins} = 0 \text{ or } 1, \;\; \forall i\in I, n\in N, s\in L. \quad (4.56)$$

In Chapter 3, we showed that constraints (4.54) are redundant for the original problem, but in the relaxed formulation they become useful to improve the bound. Note that $CM_{LR}(\pi,\eta)$ is identical to the relaxed problem

in Chapter 3, other than having c_{ijn} instead of c_{ij} and having the additional fixed term ηG in the objective function. Hence, we can apply a similar solution algorithm to $CM_{LR}(\pi, \eta)$, based on solving subproblems P_{ins} for each (i, n, s) as follows:

$$V(CM_{LR}(\pi, \eta)) = \sum_i V(P_i) + \sum_{j \in J} \pi_j + \eta G$$

and

$$V(P_i) = \min_{n \in N, s \in L} \{0, f_{is} + V(P_{ins})\},$$

where problems P_i and P_{ins} are defined for all i and (i, n, s) combinations as below: P_i:

$$\min \quad \sum_{n \in N} \sum_{s \in L} f_{is} Z_{ins} + \sum_{n \in N} \sum_{s \in L} \sum_{j \in J} \bar{c}_{ijn} X_{insj} \tag{4.57}$$

$$\text{subject to} \quad Z_{ins} \geq X_{insj} \quad \forall j \in J, n \in N, s \in L \tag{4.58}$$

$$\sum_{j \in J} d_j X_{insj} \leq \bar{\mu}_{ins} \quad \forall n \in N, s \in L \tag{4.59}$$

$$\sum_{n \in N} \sum_{s \in L} Z_{ins} \leq 1 \tag{4.60}$$

$$X_{insj} = 0 \text{ or } 1, \quad \forall n \in N, s \in L, j \in J \tag{4.61}$$

$$Z_{ins} = 0 \text{ or } 1, \quad \forall n \in N, s \in L. \tag{4.62}$$

P_{ins}:

$$\min \quad \sum_{j \in J} \bar{c}_{ijn} X_{insj} \tag{4.63}$$

$$\text{subject to} \quad \sum_{j \in J} d_j X_{insj} \leq \bar{\mu}_{ins} \tag{4.64}$$

$$X_{insj} = 0 \text{ or } 1, \quad \forall j \in J. \tag{4.65}$$

Proposition 10. *$CM_{LR}(\pi, \eta)$ is solved by finding solutions to a series of knapsack problems for each facility i, base fill rate level n, and inventory level s as P_{ins} (4.63)-(4.65).*

With Proposition 10, we develop Algorithm 4 to solve P_i. Algorithm 4 is similar to its counterpart in Chapter 2. Since each facility can serve any customer, we consider all customers having negative transportation costs c_{ijn} as candidates for assignment (as in line 3), which increases the size of the knapsack problems. Furthermore, since the transportation costs depend on the base fill rates N, we need to reorder the customers (as in line 5) for each $n \in N$ as an outer loop (in lines 2 - 32).

In lines 3-6, we find the customers that have negative transportation costs, and order them with increasing c_{ijn} values. We also initialize the set of active knapsack problems.

In lines 7-13, we calculate $V(P_{ins})$ for the stock level $s = L^-$ (which can cover demands of all customers in J^-). Then, we update the incumbent and then remove the solved knapsack problem from the active set of knapsacks.

In the loop (lines 14-31), we calculate the lower bound for each active knapsack problem, and if a knapsack problem's lower bound does not have the potential to improve the current incumbent (see the condition in line 16), we fathom these knapsacks, else we calculate their upper bounds and update the incumbent if needed. If the percentage difference between upper and lower bounds is very small (less than ϵ), we conclude that the upper bound solution

Algorithm 4 Algorithm to solve problem P_i

$UB_i^* = 0$, $(s^*, n^*) = (\emptyset, \emptyset)$ {Initialize incumbent solution}
for all $n \in N$ **do**
    $J^- = \{j \in J : \bar{c}_{ijn} < 0\}$ {Set of customers having a negative cost}
    Calculate L^-
    $J^- = \{j \in J^- : \bar{c}_{ijn[k]} \leq \bar{c}_{ijn[l]} \ \ \forall k \leq l\}$ {Order the customers with respect to their costs}
    $A = A \cup \{(n, s) : s \in L^-\}$ {Set of active knapsack problems to be solved}
    for $(s = L^-)$; $V(P_{ins}) = \sum_{j \in J^-} \bar{c}_{ijn}$ {Recall that $\mu_{ins} \geq \sum_{j \in J_{in}^-} d_j$ for $s = L^-$}
    if $f_{is} + V(P_{ins}) < UB_i^*$ **then**
        $UB_i^* = f_{is} + V(P_{ins})$ {Update incumbent solution}
        $s^* = s$ and $n^* = n$
        $\hat{J}_{ins} = J^-$ {Record the customers in the solution (i, n, s)}
    end if
    $A = A \setminus (n, s)$ {FATHOM}
    for all $(n, s) \in A$ **do**
        Calculate $LB(P_{ins})$
        if $f_{is} + LB(P_{ins}) \geq UB_i^*$ **then**
            $A = A \setminus (n, s)$ {FATHOM}
        else
            Calculate $UB(P_{ins})$
            if $(UB(P_{ins}) - LB(P_{ins}))/|LB(P_{ins})| < \epsilon$ **then**
                $UB_i^* = f_{is} + UB(P_{ins})$ {Update incumbent solution}
                $s^* = s$ and $n^* = n$
                $\hat{J}_{ins} = j \in UB(P_{ins})$ {Record the customers in the solution of (i, n, s)}
                $A = A \setminus (n, s)$ {FATHOM}
            else
                if $f_{is} + UB(P_{ins}) < UB_i^*$ **then**
                    $UB_i^* = f_{is} + UB(P_{ins})$ {Update incumbent solution}
                end if
            end if
        end if
    end for
end for
while $A \neq \emptyset$ **do**
    for all $(n, s) \in A$ **do**
        if $f_{is} + LB(P_{ins}) \geq UB_i^* - \tau$ **then**
            $A = A \setminus (n, s)$ {FATHOM}
        end if
    end for
    SOLVE $P_{i\hat{n}\hat{s}}$ for $(\hat{n}, \hat{s}) : LB(P_{i\hat{n}\hat{s}}) \leq LB(P_{ins}) \ \forall (n, s) \in A \ j \in J_{in}^-$
    $A = A \setminus (\hat{n}, \hat{s})$ {FATHOM}
    if $f_{is} + V(P_{i\hat{n}\hat{s}}) < UB_i^*$ **then**
        $UB_i^* = f_{is} + V(P_{i\hat{n}\hat{s}})$ {Update incumbent solution}
        $s^* = \hat{s}$ and $n^* = \hat{n}$
        $\hat{J}_{i\hat{n}\hat{s}} = j \in V(P_{i\hat{n}\hat{s}})$ {Record the customers in the solution of $(i, \hat{n}, \hat{s})$}
    end if
end while
if $(s^*, n^*) \neq (\emptyset, \emptyset)$ **then**
    $V(P_i) = U_i^*$
    $Z_{i,n^*,s^*} = 1$ {Open facility i with fill rate b_{in^*} and stock level s^*}
    $X_{i,n^*,s^*,j} = 1$ for all $j \in \hat{J}_{i,n^*,s^*}$ {Assign customers to facility i}
else
    Set $V(P_i) = 0$ {Do not open facility i}
end if

is in fact optimal. For the same fill rate level, while the stock level increases, we use the smaller stock level lower bound solution as a base and add new customers for the remaining capacity as we do in Chapter 3 (see Section 3.3.1.1 for details). We fathom the knapsack problems by using their lower bounds and the current incumbent.

If there are still active knapsacks, we pick a candidate problem with the lowest lower bound and solve it to optimality by using the Xpress-MP solver. Then, we update the incumbent and repeat this procedure until there are no active knapsacks left. In the final part of the algorithm (lines 47-53), we open the facility if it is profitable, i.e., the best knapsack solution value is negative.

4.3.2 Upper Bound

In Chapter 2, we showed that the integrated approach has significant benefits compared to the traditional decoupled approach DC. The DC has two sub-models solved sequentially: (1) The decoupled location and allocation model (DCL) makes facility location and customer allocation decisions with a system-based coverage constraint, and (2) the decoupled inventory model (DCI) decides the stock levels at the open facilities based on the solution of the DCL. Capitalizing on the same notation used before, we define DCL model as follows:

DCL:

$$\min \sum_{i \in I} f_i Y i + \sum_{i \in I} \sum_{j \in J} c_{ij} d_j X_{ij} \tag{4.66}$$

$$\sum_{i \in I} X_{ij} = 1, \ \ \forall j \in J \tag{4.67}$$

$$Y_i \geq X_{ij}, \ \ \forall i \in I, j \in J \tag{4.68}$$

$$\sum_{i \in I} \sum_{j \in J} \delta_{ij} d_j X_{ij} \geq G \tag{4.69}$$

$$X_{ij} = 0 \text{ or } 1, \ \forall i \in I, j \in J \tag{4.70}$$

$$Y_i = 0 \text{ or } 1, \ \forall i \in I. \tag{4.71}$$

From the *DCL* sub-model's optimal solution, we obtain $\tilde{I}$ as the set of open facilities ($\tilde{I} = \{i \in I : Y_i = 1\}$) and the demand allocation decisions $\tilde{X}_{ij}$ for all i and j. Using these, we calculate the actual mean lead time demands for all open facilities:

$$\tilde{\lambda}_i = t_i \sum_j d_j \tilde{X}_{ij}, \ \forall i \in \tilde{I}. \tag{4.72}$$

We then compute the actual fill rates for all facilities and each potential stock level:

$$\tilde{\beta}_i(s) = G_i(s-1) = \sum_{s'=0}^{s-1} \frac{(\tilde{\lambda}_i)^{s'} e^{-\tilde{\lambda}_i}}{s'!}. \tag{4.73}$$

These all become inputs into the *DCI* sub-model, which is itself a multi-facility inventory model, where the only decision variables are the stock levels, W_{is}'s. The sub-model finds the minimum-cost stock levels possible to achieve time-based service levels using the calculated actual fill rates. As the network and demand allocations are already decided and fixed, the remaining objective is to minimize the total inventory costs:

$DCI(\tilde{I}, \tilde{X}_{ij})$:

$$\min \sum_{i \in \tilde{I}} \sum_{s \in L} sh_i W_{is} \tag{4.74}$$

$$\text{subject to } \sum_{s \in L} W_{is} \geq \tilde{X}_{ij}, \ \forall i \in \tilde{I}, j \in J \tag{4.75}$$

$$\sum_{i \in \tilde{I}} \left[\left(\sum_{j \in J} \delta_{ij} d_j \tilde{X}_{ij} \right) \left(\sum_{s \in L} \tilde{\beta}_i(s) W_{is} \right) \right] \geq G \tag{4.76}$$

$$W_{is} = 0 \text{ or } 1, \ \forall i \in \tilde{I}, s \in L. \tag{4.77}$$

Since the DC approach makes decisions sequentially (first location and allocation decisions, then stock level decisions), it is not globally optimal to the OM, and it provides an upper bound, i.e., $V(DC) \geq V(OM)$.

Compared to the integrated approach, DC has the advantages of being fast and using exact fill rate values instead of using an approximate step-wise function. However, as we show in Chapter 2, the DC upper bounds may be loose, especially when stock decisions become important (cases with high stock levels and high stock costs). The weakness of the DC approach is due to the following reasons:

i While solving DCL, the appraoch is unaware of the inventory costs that the designed network may eventually require in the DCI model

ii While solving DCL, the approach is unaware of the facility fill rates that the facilities may eventually have in the DCI model

In this section, we develop a revised DC approach with a feedback mechanism based on the lower bound solution coming from CM_{LR} to overcome

the weaknesses of the DC model and to provide fast and tight upper bounds, as in Algorithm 5.

Algorithm 5 Subgradient algorithm with feedback mechanism

$LB^* = 0$
Initialize: σ
$\tilde{\beta}_i = \alpha, \ \ \forall i \in I$
$\tilde{s}_i = 1, \ \ \forall i \in I$
$\tilde{c}_{ij} = c_{ij}, \ \ \forall (i,j) \in (I,J)$
SOLVE $DCL(\tilde{\beta}_i, \tilde{s}_i, \tilde{c}_{ij}) \Longrightarrow$ OUTPUTS: $(\tilde{I}, \tilde{X}_{ij})$
Initialize: π, η from dual values of assignment and service level constraints
SOLVE $DCI(\tilde{I}, \tilde{X}_{ij})$
$UB^* = \sum_{i \in \tilde{I}} \sum_{s \in L} lh_i \tilde{W}_{is} + \sum_{i \in I} \sum_{j \in J} c_{ij} \tilde{X}_{ij}$
while Stopping conditions not satisfied **do**
    Run Algorithm 4 $\forall i \in I \Longrightarrow$ OUTPUTS: $V(P_i)$
    $LB = \sum_{i \in I} V(P_i)$
    if $LB > LB^*$ **then**
        $LB^* = LB$
    end if
    if $LB^+ \leq 2UB^*$ **then**
        $CM_{LR} \Longrightarrow$ OUTPUTS: $(\hat{I}, \hat{\beta}, \hat{s}, \hat{X}_{ij})$
        $\tilde{\beta}_i = max\{\alpha, \hat{\beta}_i\}, \ \ \forall i \in \hat{I}$
        $\tilde{\beta}_{i'} = max\{\alpha, avg_{i \in \hat{I}}\{\hat{\beta}_i\}\}, \ \ \forall i' \in I \backslash \hat{I}$
        $\tilde{s}_i = 1, \ \ \forall i \in \hat{I}$
        $\tilde{s}_{i'} = avg_{i \in \hat{I}}\{\hat{s}_i\}, \ \ \forall i' \in I \backslash \hat{I}$
        $\tilde{c}_{ij} = \nu c_{ij}, \ \ \forall (i,j) \in (\hat{I}, J : \hat{X}_{ij} = 1)$, where $\nu < 1$
        $\tilde{c}_{ij} = c_{ij}, \ \ \forall (i,j) \in (\hat{I}, J : \hat{X}_{ij} = 0)$
        SOLVE $DCL(\tilde{\beta}_i, \tilde{s}_i, \tilde{c}_{ij}) \Longrightarrow$ OUTPUTS: $(\tilde{I}, \tilde{X}_{ij})$
        SOLVE $DCI(\tilde{I}, \tilde{X}_{ij})$
        $UB^= \sum_{i \in \tilde{I}} \sum_{s \in L} sh_i \tilde{W}_{is} + \sum_{i \in I} \sum_{j \in J} c_{ij} \tilde{X}_{ij}$
        if $UB < UB^*$ **then**
            $UB^* = UB$
        end if
    end if
    if LB^* not improved in 20 consecutive iterations **then**
        $\sigma = 0.8\sigma$
    end if
    $\gamma = \sigma(UB^* - LB)/(\sum_j (1 - \sum_i \sum_n \sum_s X_{insj})^2 + (G - \sum_i \sum_n \sum_s \sum_j X_{insj})^2)$
    $\pi_j = \pi_j + \gamma(1 - \sum_i \sum_n \sum_s X_{insj}), \ \ \forall j \in J$
    $\eta = max\{0, (\eta + \gamma(G - \sum_i \sum_n \sum_s \sum_j X_{insj}))\}$
end while

The first five lines of the Algorithm 5 contain an initialization section where we set the facility fill rates to the target service level α, the stock level at the facilities to 1, and the transportation costs to the their original values. Then, we solve DCL and DCI models to obtain the initial upper

bound. Note that we initialize the Lagrangian multipliers π_j by using the dual values of the assignment constraints (4.67) and η by using the dual value of the coverage constraint (4.69). Through line 10, we start the iterative process, which contains a lower bound calculation section (lines 11 - 15), an upper bound calculation section (lines 16 - 30) and a section that updates subgradient parameters and Lagrangian multiplier values (lines 31 - 36).

Recall that the weakness of the DC approach is due to the lack of information about the effects of stock levels while designing the network. In lines 6 and 24, we solve the revised DCL model by using the feedback information obtained from the lower bound solution as follows:

i Fill rates of the facilities $\tilde{\beta}_i$: The DC approach assumes a 100% fill rate at each facility while designing the network in the DCL step, as in constraints (4.69), $\sum_{i\in I}\sum_{j\in J}(1)\delta_{ij}d_jX_{ij} \geq G$. In many cases this behavior forces DCI to keep high stock levels to satify pre-assumed high fill rates to satisfy the service level constraint (4.76), which causes the DC approach to suffer because of high stocking costs. To handle this, we incorporate the fill rates that we obtain from the lower bound solution to the DCL model, by setting fill rates of the facilities which are open in the lower bound solution ($i \in \hat{I}$), to their fill rates $\tilde{\beta}_i = \hat{\beta}_i$. For the remaining facilities ($i \in I\backslash\hat{I}$), we use the average fill rate of the open facilities. We then modify the coverage constraint to $\sum_{i\in I}\sum_{j\in J}\tilde{\beta}_i\delta_{ij}d_jX_{ij} \geq G$. Note that we force fill rates to be greater than the target service level (in lines 18 - 19 of the algorithm 5) to guarantee feasibility.

ii Stock levels at the facilities $\tilde{s}_i$: Similarly, being unaware of the facility stock levels while designing the network in DCL causes the DC approach to further suffer from high stocking costs. The DCL model only optimizes the trade-off between the facility location costs and transportation costs. For the higher fixed location costs f_i, DCL decides to open fewer facilities and pay more for the transportation (since the distance between facilities and customers will be higher). For the low fixed location costs, DCL decides to open more facilities to be closer to the customers. For the logistics network design problem, the stocking cost is a type of fixed cost, which has similar effects on the number of facilities and transportation. To incorporate the effects of the stocking costs to the DCL model, we revise fixed costs as $f_i = f_i + \tilde{s}_i h_i$, where $\tilde{s}_i$ is set to 1 for $i \in \hat{I}$ and, $\tilde{s}_i$ is set to the average stock level at the lower bound solution for $i \in I \backslash \hat{I}$ (in lines 20 - 21 of the algorithm 5). By setting stock levels at the open facilities to 1 we make these facilities preferable by aiming the DCL to provide a solution similar to the lower bound solution.

iii Allocation decisions $\tilde{X}_{ij}$: These decisions are incorporated into the DCL model to force DCL to make similar decisions with the lower bound solution. We reduce the transportation costs between the (i, j) link for $\tilde{X}_{ij} = 1$, by multiplying it with a parameter $\nu < 1$ (in lines 22 - 23 of the algorithm 5). Together with the updates in item ii., the objective function of DCL becomes $\sum_{i \in I}(f_i + \tilde{s}_i h_i)Y_i + \sum_{i \in I} \sum_{j \in J} \tilde{c}_{ij} d_j X_{ij}$.

We solve the lower bound and upper bound iteratively until we reach any of the following stopping conditions:

- When the number of iterations reaches to the maximum allowed (i.e., maximum 500 iterations; shown as "*iter*" in tables).
- When σ is too low (i.e., if $\sigma < 0.003$; shown as "*low* σ" in tables).
- When the gap between the best upper and lower bound is small (i.e., if $(UB^* - LB^*)/LB^* < 0.003$; shown as "*gap*" in tables).
- When the run time reaches the maximum limit (i.e., 3600 seconds; shown as "*time*" in tables).

4.4 Experimental Study and Results

In this section, we experiment with the approaches we developed in this chapter. Briefly, we have the following methods that can be used for the system-wide service level problem, modeled in OM:

1. Customer-centric based model CM (solved with direct MIP solver)
   - $CM(\underline{\beta})$: Provides upper bounds for OM.
   - $CM(\bar{\beta})$: Provides lower bounds for OM.
2. Relaxation and decomposition model (solved with Algorithm 5)
   - AUB: Based on the revised DC approach with a feedback mechanism providing upper bounds for OM.

– ALB: Based on $CM(\bar{\beta})$ providing the lower bound for OM.

Table 4.1 summarizes the experimental data set we use in this section. Recall that qualities of $CM(\underline{\beta})$, $CM(\bar{\beta})$ and ALB directly, and AUB indirectly (since having information from ALB) depend on the granularity of the fill rate approximation. We assume that the fill rate at an open facility should be greater than 50%, and we generate the fill rate approximation for three different granularity levels by dividing the 50% - 100% interval into 5 steps (low granularity), 10 steps (medium granularity) and 20 steps (high granularity). Since the problem size increases as the number of steps in the fill rate approximation increases, we look for high quality bounds without spending too much solution time.

We randomly assign facility and customer locations to a 100 by 100 grid and generate different size instances: a small size instance with 10 candidate facility locations and 50 customer locations (i.e., $|I| = 10$, and $|J| = 50$), a medium size instance with $|I| = 25$, and $|J| = 100$, and a rather large size instance with $|I| = 50$, and $|J| = 200$. Mean demand rates are generated uniformly with a mean of 2.5 ([from uniform distribuion [1,4]). We have a time window of 2 hours, which is assumed to be 20 units of Euclidian distance in the 100x100 grid. Each instance is run with low ($250), medium ($500), and high ($1000) holding costs. Fixed facility costs are $1000, while the transportation costs are distance-based, and the average transportation cost is around $150. We solve these instances for three different target service levels, namely 50%,

Table 4.1: Experimental data set

Fill rate granularity	5 steps, 10 steps, 20 steps
Size ($\|I\|$, $\|J\|$)	small (10, 50), medium (25, 100), large (50, 200)
Mean demand, d	uniform [1, 4]
Time window, w	2 hours
Holding cost, h	\$250, \$500, \$1000
Fixed cost, f	\$1000
Transportation cost, c	\$5 per unit distance
Target service levels, α	50%, 70%, 90%

70% and 90%. We limit the maximum solution time for any run to 3600 seconds.

We summarize the results of the small, medium and large instances in Tables 4.2, 4.3 and 4.4, respectively. Rows are categorized with respect to the fill rate granularity $|step|$, target service level α and inventory holding cost h. For referencing purposes, we name small instances as $s1 - s27$, medium instances as $m1 - m27$ and large instances as $l1 - l27$.

Columns are divided into three main sections, each separated by double-lines as follows:

The first section summarizes the results of the customer-centric based model CM; the first three columns show the lower bound value, solution time and status, and the next three columns show the upper bound value, solution time and status. The last column provides the percentage gaps between each upper and lower bound calculated as $100*(V(CM(\underline{\beta}))-V(CM(\bar{\beta})))/V(CM(\bar{\beta}))$. If an instance is solved optimally within 3600 seconds, we show its status as

"opt". If it is not finished, we show its status as "N/F".

The second section summarizes the results obtained by Algorithm 5. We show the lower and upper bound values, number of subgradient iterations performed, total solution times, and reasons for termination of the algorithm. We also show the percentage gaps between upper bound and lower bound values calculated as $100 * (AUB - ALB)/ALB$. Note that we allow 3600 seconds for each lower bound and upper bound problem of CM separately. However, Algorithm 5 provides both lower and upper bounds in 3600 seconds.

In the last section, we compare the results of CM and Algorithm 5. The gap between the lower bounds is calculated as $100*(V(CM(\bar{\beta}))-ALB)/ALB$, and the gap between the upper bounds is calculated as $100 * (v(CM(\underline{\beta})) - AUB)/AUB$. Note that if the problem $CM(\underline{\beta})$ is not solved optimally in 3600 seconds, any feasible solution still provides an upper bound (upper bound of the upper bound remains an upper bound). However, if the problem $CM(\bar{\beta})$ is not solved optimally, a feasible solution is not a valid lower bound (upper bound of the lower bound is not a lower bound). Hence, we do not compare the lower bounds of CM and the Algorithm 5 when the status of $CM(\bar{\beta})$ is "N/F".

Table 4.2: CM and Algorithm 5 results for small size problems $|I| = 10, |J| = 50$

				CM							Algorithm						Comparison	
ins	$\|step\|$	α	h ($)	$CM(\bar{\beta})$	Time	Status	$CM(\underline{\beta})$	Time	Status	Gap (%)	ALB	AUB	Iter	Time	Reason	Gap (%)	LB (%)	UB (%)
s1			250	13169	3	opt	13169	2	opt	0.0	13158	13169	72	6	gap	0.1	0.1	0.0
s2		50	500	14419	9	opt	14419	7	opt	0.0	14383	14419	107	7	gap	0.2	0.2	0.0
s3			1000	16474	106	opt	16919	100	opt	2.7	16417	17411	349	65	low σ	6.1	0.3	-2.8
s4			250	13419	25	opt	13669	27	opt	1.9	13248	13419	253	38	low σ	1.3	1.3	1.9
s5	5	70	500	14919	86	opt	15419	74	opt	3.4	14578	14919	278	46	low σ	2.3	2.3	3.4
s6			1000	17919	521	opt	18919	172	opt	5.6	17239	17919	334	58	low σ	3.9	3.9	5.6
s7			250	14169	21	opt	16616	4	opt	17.3	14037	14669	443	66	low σ	4.5	0.9	13.3
s8		90	500	16419	96	opt	19866	6	opt	21.0	16100	17419	340	59	low σ	8.2	2.0	14.1
s9			1000	20919	121	opt	26366	7	opt	26.0	20006	22481	473	134	low σ	12.4	4.6	17.3
s10			250	13169	15	opt	13169	14	opt	0.0	13148	13169	62	9	gap	0.2	0.2	0.0
s11		50	500	14419	24	opt	14419	25	opt	0.0	14384	14419	93	11	gap	0.2	0.2	0.0
s12			1000	16507	246	opt	16919	2979	opt	2.5	16441	16961	314	65	low σ	3.2	0.4	-0.2
s13			250	13419	27	opt	13419	21	opt	0.0	13297	13419	293	55	low σ	0.9	0.9	0.0
s14	10	70	500	14919	136	opt	14919	35	opt	0.0	14677	14919	364	80	low σ	1.6	1.6	0.0
s15			1000	17919	971	opt	17919	48	opt	0.0	17387	17919	263	60	low σ	3.1	3.1	0.0
s16			250	14419	515	opt	14919	105	opt	3.5	14203	14669	267	50	low σ	3.3	1.5	1.7
s17		90	500	16919	1146	opt	17919	391	opt	5.9	16430	17419	347	97	low σ	6.0	3.0	2.9
s18			1000	21531	1188	opt	23468	377	opt	9.0	20429	22468	345	109	low σ	10.0	5.4	4.5
s19			250	13169	88	opt	13169	88	opt	0.0	13140	13169	45	14	gap	0.2	0.2	0.0
s20		50	500	14419	131	opt	14419	123	opt	0.0	14382	14419	67	19	gap	0.3	0.3	0.0
s21			1000	17261	3624	N/F	17081	3622	N/F		16476	16961	428	163	low σ	2.9		0.7
s22			250	13419	123	opt	13419	128	opt	0.0	13341	13419	316	219	low σ	0.6	0.6	0.0
s23	20	70	500	14919	391	opt	14919	187	opt	0.0	14752	14919	315	159	low σ	1.1	1.1	0.0
s24			1000	17919	2166	opt	17919	355	opt	0.0	17570	17919	375	209	low σ	2.0	2.0	0.0
s25			250	14419	289	opt	14669	225	opt	1.7	14362	14669	325	161	low σ	2.1	0.4	0.0
s26		90	500	16919	3028	opt	17419	578	opt	3.0	16675	17419	414	221	low σ	4.5	1.5	0.0
s27			1000	21919	3373	opt	22953	3623	N/F	4.7	20781	22468	362	263	low σ	8.1	5.5	2.2

Table 4.2 summarizes the results for the small instances with 10 facilities and 50 customers. For the low fill rate granularity ($|step| = 5$, instances $s1 - s9$), both CM and Algorithm 5 bounds are loose since the average gap between the lower and upper bounds for CM is 8.6%, and 4.3% for Algorithm 5. The improvement of Algorithm 5 over CM in the gaps is mainly due to its having better upper bounds. As we see in the comparison section, other than instance $s3$, Algorithm 5 provides better upper bounds than $CM(\underline{\beta})$, and the average improvement is 5.8%. Recall that due to relaxation, ALB is a lower bound on $CM(\bar{\beta})$, and the average gap between ALB and $V(CM(\bar{\beta}))$ is 1.7%. The average time spent performing ALB and AUB calculations is about 53 seconds, whereas the average time spent in CM lower bounds is 110 seconds and the average time spent in CM upper bounds is 44 seconds.

For medium level granularity ($|step| = 10$, instances $s10 - s18$), as expected both CM bounds (average gap 2.3%) and Algorithm 5 bounds (average gap 3.2%) are improved. While the average solution time of Algorithm 5 stays almost the same (59 seconds), the CM average solution time increases significantly to 450 seconds for each of the lower and upper bounds (total about 900 seconds per instance).

When the fill rate granularity level is high ($|step| = 20$, instances $s19 - s27$), in general both methods' bounds improve. The average gap between CM bounds is 1.2% (ignoring the invalid lower bound for instance $s21$, see the discussion below), and the average gap between Algorithm 5 bounds is 2.4%. Note that when the CM bounds come from optimal solutions, we have

the following relations:

$$V(CM(\bar{\beta}, |step| = 5)) \leq V(CM(\bar{\beta}, |step| = 10)) \leq V(CM(\bar{\beta}, |step| = 20))$$

$$V(CM(\underline{\beta}, |step| = 5)) \geq V(CM(\underline{\beta}, |step| = 10)) \geq V(CM(\underline{\beta}, |step| = 20))$$

However, when the solutions are not optimal, we cannot guarantee these relations. For example, the upper bound for the instance with a 50% target service level and a 1000$ holding cost is solved to optimality when the fill rate granularity is low ($s3$) and medium ($s12$), hence the relation $V(CM(\underline{\beta}, |step| = 5)) = 16919 \geq V(CM(\underline{\beta}, |step| = 10)) = 16919$ holds. When the fill rate granularity is high ($s21$), however, the solution is not optimal and $V(CM(\underline{\beta}, |step| = 20)) = 17081$, which is greater than $V(CM(\underline{\beta}, |step| = 10))$.

For the small instances we can conclude that both CM and Algorithm 5 provide reasonable bounds. With the low fill rate granularity, Algorithm 5 upper bounds are significantly better than the CM upper bounds, and while the fill rate granularity increases, both upper bounds get closer. CM spends more time than Algorithm 5 in all instances, and it reaches the time limit for some instances when the fill rate granularity is high.

Table 4.3: CM and Algorithm 5 results for medium size problems $|I| = 25, |J| = 100$

				CM							Algorithm						Comparison	
ins	$\|step\|$	α	h ($)	$CM(\bar{\beta})$	Time	Status	$CM(\underline{\beta})$	Time	Status	Gap (%)	ALB	AUB	Iter	Time	Reason	Gap (%)	LB (%)	UB (%)
m1			250	23604	175	opt	23778	815	opt	0.7	23601	23767	381	429	low σ	0.7	0.0	0.0
m2		50	500	25354	383	opt	25778	3618	N/F	1.7	25304	25767	500	267	iter	1.8	0.2	0.0
m3			1000	30032	3619	N/F	30049	3618	N/F		28405	29767	500	450	iter	4.8		0.9
m4			250	24267	3621	N/F	24818	1105	opt		24096	24698	500	427	iter	2.5		0.5
m5	5	70	500	27324	3617	N/F	28341	3620	N/F		26341	27448	400	270	low σ	4.2		3.3
m6			1000	32252	3618	N/F	34394	3623	N/F		30760	32960	375	392	low σ	7.2		4.4
m7			250	26265	3624	N/F	29475	319	opt		26106	27225	500	1080	iter	4.3		8.3
m8		90	500	29765	1554	opt	34725	1792	opt	16.7	29438	31725	334	577	low σ	7.8	1.1	9.5
m9			1000	36765	3617	N/F	44770	1923	opt		35895	40725	419	926	low σ	13.5		9.9
m10			250	23604	854	opt	24530	3633	N/F	3.9	23601	23767	364	513	low σ	0.7	0.0	3.2
m11		50	500	27604	3629	N/F	26602	3630	N/F		25304	25767	500	781	iter	1.8		3.2
m12			1000	31224	3625	N/F	30175	3629	N/F		28514	28947	467	1058	low σ	1.5		4.2
m13			250	25129	3631	N/F	24784	3629	N/F		24314	24698	413	663	low σ	1.6		0.3
m14	10	70	500	27846	3631	N/F	28449	3625	N/F		26749	27448	367	588	low σ	2.6		3.6
m15			1000	33388	3630	N/F	33217	3628	N/F		31606	32948	500	824	iter	4.2		0.8
m16			250	27459	3633	N/F	27475	3624	N/F		26429	27225	500	1349	iter	3.0		0.9
m17		90	500	30836	3625	N/F	32225	3639	N/F		30179	31741	422	684	low σ	5.2		1.5
m18			1000	39716	3627	N/F	42548	3629	N/F		37022	41108	500	1126	iter	11.0		3.5
m19			250	29517	3626	N/F	30506	3633	N/F		23598	23767	447	779	low σ	0.7		28.4
m20		50	500	30767	3628	N/F	33767	3643	N/F		25313	25767	347	450	low σ	1.8		31.0
m21			1000	57520	3626	N/F	45146	3627	N/F		28700	28958	500	1195	iter	0.9		55.9
m22			250	45752	3623	N/F	47269	3620	N/F		24434	24698	500	1456	iter	1.1		91.4
m23	20	70	500	51757	3617	N/F	62267	3614	N/F		27019	27448	500	1758	iter	1.6		126.9
m24			1000		5723	N/F		5610	N/F		32116	32948	495	1591	low σ	2.6		
m25			250	60672	3620	N/F	63803	3619	N/F		26739	27225	500	1896	iter	1.8		134.4
m26		90	500	75301	3619	N/F	60132	3626	N/F		30728	31725	500	1100	iter	3.2		89.5
m27			1000		3893	N/F		4358	N/F		37702	40917	500	1266	iter	8.5		

Table 4.3 summarizes the results for the medium instances with 25 facilities and 100 customers. Other than in a few instances, CM is unable to provide optimal lower and upper bounds, hence we can only calculate the gap for instances $m1, m2, m8$ and $m10$. Recall that when the solutions are not optimal, any feasible solution for upper bounds $CM(\underline{\beta})$ are still valid. When the fill rate granularity level is increased from 5 to 10, while some $CM(\underline{\beta})$ values improve (such as $m13$, $m15 - m19$), many of them get worse because of the time limitation. For the high fill rate granularity, all of the $CM(\underline{\beta})$ values get worse, and in two instances ($m24$ and $m27$) we do not have any feasible solutions.

For medium size instances, Algorithm 5 is still able to provide lower and upper bounds without any time restriction, and even for the high fill rate granularity, the average time spent per an instance is 1270 seconds. While the fill rate granularity increases, the percentage gap between AUB and ALB improves: 5.3% for low granularity, 3.5% for medium granularity and 2.5% for high granularity. When we have optimal $CM(\bar{\beta})$ solution, we see that the gap between ALB and $V(CM(\bar{\beta}))$ is very low. We observe that for any setting, AUB improves $CM(\underline{\beta})$ solutions. This improvement is significant, especially when the fill rate granularity is high.

Table 4.4: Algorithm 5 results for large size problems $|I| = 50, |J| = 200$

				Algorithm					
ins	$\|step\|$	α	h ($)	ALB	AUB	Iter	Time	Reason	Gap (%)
l1			250	42639	42677	105	182	gap	0.1
l2		50	500	46075	46152	106	103	gap	0.2
l3			1000	52636	52718	70	91	gap	0.2
l4			250	43565	44177	500	1971	iter	1.4
l5	5	70	500	47550	48928	411	1336	low σ	2.9
l6			1000	55399	58445	477	2163	low σ	5.5
l7			250	45652	47195	500	1924	iter	3.4
l8		90	500	51624	54928	489	2234	low σ	6.4
l9			1000	63248	70068	500	2151	iter	10.8
l10			250	42550	42677	105	339	gap	0.3
l11		50	500	46069	46152	108	240	gap	0.2
l12			1000	52564	52716	66	155	gap	0.3
l13			250	43793	44177	500	2451	iter	0.9
l14	10	70	500	48078	48928	500	2575	iter	1.8
l15			1000	56568	58439	500	2780	iter	3.3
l16			250	46268	47205	500	3149	iter	2.0
l17		90	500	52917	55022	500	2843	iter	4.0
l18			1000	65592	70143	500	2061	iter	6.9
l19			250	42612	42677	92	852	gap	0.2
l20		50	500	45821	45927	102	554	gap	0.2
l21			1000	52585	52716	96	498	gap	0.3
l22			250	43942	44177	500	2940	iter	0.5
l23	20	70	500	48417	48927	499	3640	time	1.1
l24			1000	57271	58467	454	3620	time	2.1
l25			250	46611	47204	428	3612	time	1.3
l26		90	500	53584	55079	438	3606	time	2.8
l27			1000	66833	70749	500	3400	iter	5.9

Table 4.4 summarizes the results for the large instances with 50 facilities and 200 customers. For these instances, CM is totally unable to provide any feasible solution within the given time limit. Therefore, we only show Algorithm 5 results in this table. We see that the gap between AUB and ALB is still small for these instances, and it improves with increasing fill rate granularity.

4.5 Conclusion

In this chapter, we apply the successful methodology of the customer-centric problem developed in the Chapter 3 to the system-wide problem. We revise the MIP formulation of the customer-centric problem to provide lower and upper bounds for the original system-wide problem. Based on this new formulation, we develop a relaxation and decomposition based algorithm which is capable to solve large instances of up to 50 facilities and 200 customers with medium demands with a mean of 2.5 units.

Chapter 5

Column Generation Extension

5.1 Column Generation for the Integrated Problem

5.1.1 A New Formulation

As part of our investigation towards developing new solution methodologies for the integrated model, we plan to utilize column generation. Column generation for integer programming problems is based on Dantzig-Wolfe decomposition originally developed to solve large linear programming problems by Dantzig and Wolfe (1960). The idea is to decompose the overall problem into smaller subproblems and a so-called master problem that controls the interaction between these subproblems.

Although the solution of large-scale LP problems does not typically require the use of DW decomposition anymore, the technique has been successfully used to solve integer programming problems. DW-based column generation combined with branch-and-bound to solve IP problems is called branch-and-price. Barnhart et al. (1998) and Wilhelm (2001) review the recent literature on column generation and branch-and-price for IP.

The steps of a generic column generation algorithm are as follows (Bertsimas and Tsitsiklis (1997)):

- *Formulation of the original problem, master problem and pricing problem*
- Step 0: Initialize the restricted master problem with a feasible solution
- Step 1: Solve the restricted master problem to obtain dual values of decoupling constraints.
- Step 2: Solve the subproblems
- Step 3: Check for termination (nonnegative reduced costs for a minimization problem)
- Step 4: Augment new columns to the restricted master problem, and go to 1.

Formulation is a critical precursor to the algorithm as it is an important determinant of success in solving the underlying problem. One prefers easily solvable subproblems after decomposition, with a master problem that will lead to tighter bounds than a normal LP relaxation. Hence, if a problem can be formulated in different ways, which is the case for most IP problems, we try to identify one that is amenable to a successful column generation implementation.

Barnhart et al. (2000) model an integer multicommodity network flow problem using "path formulation" instead of traditional "node-arc formulation." The path formulation has far more variables and fewer constraints than the node-arc formulation, and the new formulation has shortest path subproblems, which could be efficiently solved. Barnhart and Cohn (2002) model a

coverage-based service parts logistics problem where each facility stocks one unit. Instead of using traditional assignment variables X_{il} (1 if warehouse i stocks part l, 0 otherwise), they use "composite variables" X_{gl} (1 if warehouse group g stocks part l, 0 otherwise); where the group g is a subset of warehouses that provides full coverage for l. The new formulation greatly reduces the number of constraints, but significantly increases the number of binary variables. The master problem is a set covering problem, which is known to have very tight LP relaxation bounds.

The benefits of these formulations are explained in Barnhart et al. (1998) as follows:

- A compact formulation of a MIP may have a weak LP relaxation
- A compact formulation of a MIP may have a symmetric structure that causes branch-and-bound to perform poorly because the problem barely changes after branching
- A formulation with a huge number of variables may result in better subproblems.

Hence, we reformulate the integrated problem introduced in Section 2.2. In the previous formulation of the integrated logistics network design and inventory model, we use three types of assignment variables for each facility (i) and part (k) pair: (1) X_{ijk}: assignment of customers (j), (2) W_{ikl}: assignment of stock levels (l), and (3) Q_{ikn}: assignment of mean lead time demand intervals

in the stepwise fill rate function (n). Here, the main idea is to use one variable Z_{ijknl} to make all of the assignments together. We first introduce the new formulation and then detail a column generation based decomposition along with our preliminary results.

Using the same notation introduced earlier, the new model is as follows:

$$\min \sum_{i\in I} f_i Yi + \sum_{i\in I}\sum_{k\in K}\sum_{l\in L} lh_{ik} W_{ikl} + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{n\in N}\sum_{l\in L} c_{ijk} d_{jk} Z_{ijknl} \quad (5.1)$$

$$\sum_{i\in I}\sum_{n\in N}\sum_{l\in L} Z_{ijknl} = 1,\ \forall j,k \quad (5.2)$$

$$W_{ikl} \geq \sum_{n\in N} Z_{ijknl},\ \forall i,j,k,l \quad (5.3)$$

$$Q_{ikn} \geq \sum_{l\in L} Z_{ijknl},\ \forall i,j,k,n \quad (5.4)$$

$$Y_i \geq \sum_{l\in L} W_{ikl},\ \forall i,k \quad (5.5)$$

$$1 \geq \sum_{n\in N} Q_{ikn},\ \forall i,k \quad (5.6)$$

$$a_{k,n-1} Q_{ikn} \leq t_{ik} \sum_{j\in J}\sum_{l\in L} d_{jk} Z_{ijknl},\ \forall i,k,n \quad (5.7)$$

$$\sum_{i\in I}\sum_{j\in J}\sum_{n\in N}\sum_{l\in L} (b_{kln}\delta_{ij} d_{jk}) Z_{ijknl} \geq \alpha_k d_k,\ \forall k \quad (5.8)$$

$$Y_i, W_{ikl}, Q_{ikn}, Z_{ijknl} = 0 \text{ or } 1,\ \forall i,j,k,l,n. \quad (5.9)$$

Constraints (5.1, 5.2, 5.3, and 5.5) are similar to the constraints (2.39-2.42) in the previous formulation. Here, instead of the table look-up constraints (2.43-2.47), we now use constraints (5.4) and (5.7); and finally, instead of the service level constraints (2.48-2.53), we now use constraints (5.8).

Table 5.1: Number of variables and constraints for old and new models

	Number of variables	Number of constraints
Old Model	$I(1+K(L+N+N+LN+J))$	$K+JK+IK(J+3+3N+5LN)$
New Model	$I(1+K(L+N+JLN))$	$K+JK+IK(JL+JN+2+N)$

Table 5.1 lists the numbers of variables and constraints for both formulations. The old formulation has variables in the order of $O(IK(J+LN))$ whereas the new formulation in the order of $O(IJKLN)$. The number of constraints is in the order of $O(IK(J+LN))$ for the old formulation and $O(IKJ(L+N))$ for the new formulation. For example, with 10 facilities, 20 customers, 2 parts, 5 stock levels, and 5 fill rate function steps, the old formulation has 1210 variables and 3302 constraints; and the new formulation has 10210 variables and 4182 constraints. For a larger size problem, the difference is more magnified: A problem with 25 facilities, 150 customers, 10 parts, 5 stock levels and 10 fill rate function steps, the old formulation has 55025 variables and 106010 constraints; and the new one 1878775 variables and 565760 constraints.

Table 5.2 shows the results of a preliminary comparison between the two formulations, both solved directly in a branch-and-bound based method used in a commercial solver (Xpress-MP by Dash Optimization is used here). The table summarizes the results for a problem instance with 45 facilities, 90 customers, 2 parts, 6 stock levels and 10 fill rate function steps, with varying target service levels ($\alpha = 0.1, 0.3, 0.5$, and 0.7). We run the two models for

Table 5.2: Comparison of the old and new formulations for an instance with $|I| = 45$, $|J| = 90$, $|K| = 2$, $|L| = 6$ and $|N| = 10$

		New Model			Old Model		
α		IP	LP	% Gap	IP	LP	% Gap
0.1	obj	235487	235385.6	0.043078	*235487*	234511.9	0.414063
	time	1319.59	379.031		7241.53	14.078	
0.3	obj	*237877*	237693.2	0.07725	*NA*	234513.9	1.413805
	time	7222.84	502.938		7235.05	12.313	
0.5	obj	*241207*	240645.1	0.232941	*NA*	234515.9	2.774007
	time	7225	715.328		7245.8	12.891	
0.7	obj	*244690*	243915.1	0.31667	*NA*	234518	4.15708
	time	7214.38	676.813		7232.66	13.188	

7200 seconds, and report the values of the best integer solutions (if found), *initial* LP relaxation bounds, and the percentage gaps between the LP lower bounds and integer solutions. An italic value is the best reported integer solution, not proven to be optimal within 7200 seconds. A cell with *NA* shows that the algorithm could not find an integer feasible solution in the given time limit. Note that the new model's LP relaxations take longer to solve, but they provide much tighter bounds and help produce integer feasible solutions with very small percentage gaps, all less than 0.5%. In the next section, we tackle the solution of the LP relaxations using column generation.

5.1.2 Column Generation for the New Formulation

The idea behind the proposed column generation is to decompose the new formulation by creating subproblems for each facility. Note that the con-

straints (5.3, 5.4, 5.5, 5.6) and (5.7) involve each facility independently. Hence, we need the demand satisfaction constraints (5.2), and service level constraints (5.8) in the master problem, as they tie the facilities together.

Master Problem

The master problem has the convexity constraints along with the service level and demand satisfaction constraints. It involves decision variables λ^{ri} that takes a value between 0 and 1 if column r is used for facility i. R is the set of all columns, partitioned by R^i that lists the columns for each facility i. The columns are represented by parameters z^{ri}_{jknl}, and w^{ri}_{kl}. Note that a column represents a feasible set of assignments for a single facility and its stock level. The following is the master problem:

$$\min \sum_{i\in I}\sum_{r\in R^i}\left(f^i + \sum_{k\in K}\sum_{l\in L} h^i_{kl} w^{ri}_{kl} + \sum_{j\in J}\sum_{k\in K}\sum_{n\in N}\sum_{l\in L} c^i_{jk} d_{jk} z^{ri}_{jknl}\right)\lambda^{ri} \quad (5.10)$$

$$\sum_{r\in R^i} \lambda^{ri} \leq 1 \ \forall i \quad \Longrightarrow \gamma_i \quad (5.11)$$

$$\sum_{i\in I}\sum_{r\in R^i}\left(\sum_{n\in N}\sum_{l\in L} z^{ri}_{jknl}\right)\lambda^{ri} = 1 \ \forall j,k \quad \Longrightarrow \pi_{jk} \quad (5.12)$$

$$\sum_{i\in I}\sum_{r\in R^i}\left(\sum_{j\in J}\sum_{n\in N}\sum_{l\in L}(b_{knl}\delta^i_j d_{jk}) z^{ri}_{jknl}\right)\lambda^{ri} \geq \alpha_k d_k \ \forall k \quad \Longrightarrow \sigma_k \quad (5.13)$$

$$0 \leq \lambda^{ri} \leq 1 \ \forall r,i. \quad (5.14)$$

The coefficients of variables λ^{ri} in the objective are the total costs of opening facility i and making the assignments and using the stock levels represented

in column r for facility i. Constraints (5.11) guarantees that each facility not to be open more than once. Constraints (5.12) are demand satisfaction constraints and constraints (5.13) are service level constraints, both written across columns. These three sets of constraints (5.11, 5.12, and 5.13) have associated dual variables γ_i, π_{jk}, and σ_k, which will be used in the pricing subproblems (They are shown next to the constraints in the above formulation).

We start solving the master problem with a feasible set of solutions (columns) that can be obtained in different ways: Solve a phase I problem using a dummy facility (assign all customers to a super dummy facility which can satisfy required service with very high costs), or use a heuristic technique such as the decoupled approach we described earlier. Ideally, starting with good feasible solutions should speed up the algorithm.

We need optimal values of the dual variables γ_i, π_{jk}, σ_k from the master problem to calculate minimum reduced costs and to find any improving columns if exist. For this purpose, we solve the pricing subproblems.

Pricing Problems

In the pricing problem, we minimize the reduced cost associated with each facility. We first list the pricing problem formulation for facility i:

$$\min \left[\left(f^i - \gamma_i\right) + \sum_{k\in K}\sum_{l\in L} h^i_{kl} W^i_{kl} + \sum_{j\in J}\sum_{k\in K}\sum_{n\in N}\sum_{l\in L} \left(c^i_{jk} d_{jk} - \pi_{jk} - \sigma_k b_{knl} \delta^i_j d_{jk}\right) Z^i_{jknl} \right] \tag{5.15}$$

$$= \min \left[\bar{f}^i + \sum_{k \in K} \sum_{l \in L} h^i_{kl} W^i_{kl} + \sum_{j \in J} \sum_{k \in K} \sum_{n \in N} \sum_{l \in L} \bar{c}^i_{jknl} Z^i_{jknl} \right] \tag{5.16}$$

$$W^i_{kl} \geq \sum_{n \in N} Z^i_{jknl}, \ \forall j, k, l \tag{5.17}$$

$$Q^i_{kn} \geq \sum_{l \in L} Z^i_{jknl}, \ \forall j, k, n \tag{5.18}$$

$$1 \geq \sum_{l \in L} W^i_{kl}, \ \forall k \tag{5.19}$$

$$1 \geq \sum_{n \in N} Q^i_{kn}, \ \forall k \tag{5.20}$$

$$a_{k,n-1} Q^i_{kn} \leq t^i_k \sum_{j \in J} \sum_{l \in L} d_{jk} Z^i_{jknl}, \ \forall k, n \tag{5.21}$$

$$Z^i_{jknl}, W^i_{kl}, Q^i_{kn} = 0 \text{ or } 1, \ \forall j, k, l, n. \tag{5.22}$$

In the objective (5.16), the fixed cost is written as $\bar{f}^i$ which is equal to $(f^i - \gamma_i)$. Since γ_i is nonpositive, the new fixed cost is always positive. Similarly, the transportation costs are written as $\bar{c}^i_{jkln}$, which is set as $\bar{c}^i_{jkln} = \left(c^i_{jk} d_{jk} - \pi_{jk} - \sigma_k b_{kln} \delta^i_j d_{jk}\right)$, where i is used as superscript for consistency only (e.g., $c^i_{jk} = c_{ijk}$ in the original formulation). (The same notational convention is used in the rest of the model, for example, W_{ikl} in the original formulation is now W^i_{kl}). Here, π_{jk} is unrestricted in sign, and σ_k is nonnegative; $\bar{c}^i_{jkln}$ can be positive, 0, or negative.

Note that a positive σ_k encourages the objective function to choose larger b_{kln} values to promote service satisfaction, which happens to be with low n (demand) and high l (stock level) values. On the other hand, the inventory

Table 5.3: Comparison of old, new and column generation formulations in terms of their initial LP, and optimal IP solution values and times for $|I| = 4$, $|J| = 8$, $|K| = 2$, $|L| = 6$ and $|N| = 10$

		New Model			Old Model			Column Gen.	
α		IP	LP	% Gap	IP	LP	% Gap	LP	gap
0.1	obj	26346	23246	11.766	26346	21880.5	16.949	23273.1	11.663
	time	0.344	0.203		6.188	0.093		30.578	
0.3	obj	26346	26023.2	1.225	26346	21881.4	16.946	26059.3	1.088
	time	0.328	0.203		3.094	0.109		28.297	
0.5	obj	32191	29586.5	8.090	32191	21882.3	32.023	29678.8	7.804
	time	0.453	0.312		67.312	0.079		41.281	
0.7	obj	38136	35773.1	6.195	38136	21883.3	42.617	36051.7	5.465
	time	0.422	0.187		113.438	0.094		39.671	

cost term $h^i_{kl}W^i_{kl}$ tries to choose a low l value. With constraints (5.21), the model finds the optimal n and l combinations for each part k.

Column Generation Preliminary Results

Table 5.3 compares the old model, new model and column generation for a small problem instance with 4 facilities, 8 customers, 2 parts, 6 stock levels, and 10 fill rate function steps. All service levels are solved to optimality with the old and new formulations, due to the size of the problem instance. The column generation LP produces tighter bounds at all service levels than any other LP. However, the computation times for column generation are not competitive with those of the old and new model's LP relaxations. We think that column generation can be improved by finding more efficient ways to generate high quality columns.

To this end, we leave the investigatigation of finding ways to improve the column generation method, especially in its computation time as a future research item. We believe that alternative formulations and decompositions which will potentially lead to different master and pricing problems may be necessary.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

In this dissertation, we consider an integrated approach to model both logistic network design and inventory stocking decisions with time-based service requirements, all in one monolithic formulation. The overall challenge in this research is to capture the interactions between these two decision sets so that the decisions are compatible and globally optimal. We conduct extensive computational tests to quantify the benefits of considering inventory explicitly as part of the network design model using data sets based on the real-life industrial data.

We observe that we can achieve the same service levels with less cost with the integrated approach when compared with the traditional approach of making these decisions sequentially. Similarly, higher service levels can be achieved for a given budget with the integrated approach. We gain increasingly significant benefits from integration when inventory decisions become more dominant (i.e., the problems with high inventory holding costs caused by expensive parts, higher inventory levels due to longer time windows, or higher required service levels). This is intuitive since the decoupled approach

ignores the inventory decisions while designing the network, and when these decisions become more important and interact with network design decisions more, the cost of ignoring them gets higher.

For larger, more realistic size instances, the need for more efficient solution techniques is undeniable. To achieve this, we focus on smaller pieces of the original problem, to gain intuition and develop alternative methodologies. The customer-centric service level problem requires each customer's target service level to be fully satisfied, which enables us to decompose overall problem into subproblems, one for each facility, after relaxing the customer assignment constraints. We show that the customer-centric problem can be formulated as a version of the capacitated facility location problem. We develop a hybrid/Lagrangian relaxation based algorithm, enhanced with improvement techniques, which can be applicable to large instances (up to 100 facilities and 400 customers).

We successfully implement the methodology and insights we learned from the customer-centric problem to the original system-wide service level problem. We show that we can obtain high quality lower and upper bounds for the system-wide service level problems by solving series of knapsack problems. We succeed to solve large instances with medium-level demands (up to mean demand of 4 units per customer).

6.2 Future Research

To this end, we recommend the following potential extensions for future research:

- Extensions of the integrated model
  - Consider an extension of the approach to model multiple regions, each with region-based service levels, and/or multi-tiered service time windows. This extension is useful when considering systems with customers having products with different criticality levels. Depending on the characteristics of the customers in a region (their service contracts and overall density), a potentially different service level can be defined for each region. To incorporate them into the integrated model, we need service level constraints for each region, part, and time window. (We can use α_{krw} to denote the target service level for part k in region r for time window w).
  - The real SPL problems involve many other complexities. Some of these can be considered in an extended version of the integrated model. Such extensions for SPL-based problems include product-level service levels along with part commonality (Jeet and Kutanoglu (2005), Jeet (2006)), inventory sharing among facilities, and multi-echelon versions where facilities at two or more echelons are located.

- Extend the integrated model for "high level" demand cases. Even though we consider the "low" and "medium" demand case (motivated by SPL applications), it can be easily extended to "high level" demand cases where the ordering is by cases/containers. The main need here is to consider (Q, r) models (instead of one-for-one replenishment) where an EOQ-type approximation can be used for the order quantity (Q) and the fill rates can be calculated similarly. This should work as long as the quantities in cases/containers is small.

- Other Analyses

  - Completing the methodological development of Chapter 5 based on decomposition and column generation. Recall that the proposed methodology for the column generation in Chapter 5 is promising in terms of providing tight lower bounds, compared to the LP bounds. However, our column generation approach takes too much time. We believe that, with different formulations (similar to the one in Chapter 4) and heuristics for the pricing problems, the column generation solution times and results can be improved.

  - Extensively test these models and solution approaches using realistic, industry-scale data (such as available national and global IBM data). Provide an industrial case study. Compare our results with IBM service parts solutions' results, if available.

Appendix

Example for Inventory Reserving for Critical Customers

Assume that facility i serves two customers, a critical customer j_1 requiring $\alpha_{j_1} = 90\%$ service with annual mean demand $d_{j_1} = 5$ units, and a non-critical customer j_2 requiring $\alpha_{j_2} = 50\%$ service with annual mean demand $d_{j_2} = 30$ units.

According to Proposition 3, $\beta_i \geq \max\{\alpha_{j_1}, \alpha_{j_2}\}$, base fill rate β_i at facility i should be 90%.

Without any prioritization, to guarantee 90% fill rate with total mean demand of $d_{j_1} + d_{j_2} = 35$, facility i should keep $S_i = 3$ units in the stock (assuming that lead time is one week). Note that this stock level over-satisfies the service level requirement of non-critical customer j_2.

However, facility i may reserve 1 unit of stock for the critical customer j_1 (which guarantees 90% fill rate for mean demand of $d_{j_1} = 5$), and keep another unit stock for all customers j_1 and j_2 (which guarantees 50% fill rate for mean demand of $d_{j_1} + d_{j_2} = 35$), with overall stock level $S_i = 2$.

Bibliography

V. Agrawal and S. Seshadri. Distribution free bounds for service constrained (q,r) inventory systems. *Naval Research Logistics*, 47:636–656, 2000.

P. Alfredsson and J. Verrijdt. Modeling emergency supply flexibility in a two-echelon inventory systems. *Management Science*, 45(10):1416–1431, 1999.

F. Barahona and D. Jensen. Plant location with minimum inventory. *Mathematical Programming*, 83:101–111, 1998.

C. Barnhart and A. M. Cohn. Composite-variable models for service parts logistics. Technical report, Massacusets Institute of Technology, 2004.

C. Barnhart and A. M. Cohn. Composite variable models for service parts logistics. *Submitted to Annals of Operations Research*, 2002.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch and price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

C. Barnhart, C. A. Hane, and P. H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, Massachusetts, 1997.

M. Bundschuh, D. Klabjan, and D.L. Thurston. Modeling robust and reliable supply chains. *Optimization Online*, 2003.

F.Y. Chen and D. Krass. Inventory models with minimal service level constraints. *European Journal of Operational Research*, 134:120–140, 2001.

M.A. Cohen, P.R. Kleindorfer, and H.L. Lee. Optimal stocking policies for low usage items in multi-echelon inventory systems. *Naval Research Logistics Quarterly*, 33:17–38, 1986.

M.A. Cohen, P.R. Kleindorfer, and H.L. Lee. Service constrained (s,S) inventory systems with priority demand classes and lost sales. *Management Science*, 34(4):482–499, 1988.

M.A. Cohen, P.R. Kleindorfer, and H.L. Lee. Near-optimal service constrained stocking policies for spare parts. *Operations Research*, 37(1):104–117, 1989.

M.A. Cohen, P. Kamesam, P.R. Kleindorfer, H.L. Lee, and A. Tekerian. OPTIMIZER: IBM's multi-echelon inventory system for managing service logistics. *Interfaces*, 20(1):65–82, 1990.

M.A. Cohen, P.R. Kleindorfer, H.L. Lee, and D.F. Pyke. Multi-item service constrained (s,S) policies for spare parts logistics systems. *Naval Research Logistics*, 39:561–577, 1992.

M.A. Cohen, Y-S. Zheng, and V. Agrawal. Service parts logistics: A benchmark analysis. *IIE Transactions*, 29:627–639, 1997.

M.A. Cohen, Y-S. Zheng, and Y. Wang. Identifying opportunities for improving teradyne's service parts logistics system. *Interfaces*, 29(4):1–18, 1999.

M.A. Cohen, C. Cull, H.L. Lee, and D. Willen. Saturn's supply chain innovation: High value in after sales service. *MIT Sloan Management Review*, 41 (4, Summer):93–101, 2000.

G.B. Dantzig and P. Wolfe. The decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

M.S. Daskin. *Network and Discrete Location: Models, Algorithms and Methods*. Wiley-Interscience, New York, 1995.

M.S. Daskin, C.R. Coullard, and Z.-J.M. Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, 110:13–106, 2002.

M.S. Daskin, L.V. Snyder, and R.T. Berger. Facility location in supply chain design. Technical report, Lehigh University, December 2003.

Z. Drezner. *Facility Location: A survey of applications and methods*. Springer, New York, NY, 1995.

S.J. Erlebacher and R.D. Meller. The interaction of location and inventory in designing distribution systems. *IIE Transactions*, 32:155–166, 2002.

Wallace J. Hopp and Mark L. Spearman. *Factory Physics*. Irwin Professional, 2000.

V. Jeet and E. Kutanoglu. Logistics network design with inventory stocking, time-based service allocation and part commonality. Technical report, The University of Texas at Austin, May 2005.

Vishv Jeet. *Logistics Network Design with Inventory Stocking, Time-based Service and Part Commonality*. PhD thesis, The University of Texas at Austin, 2006.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, August 2005.

T.L. Magnanti and R.T. Wong. Network design and transportation planning: Model and algorithms. *Transportation Science*, 18:1–55, 1984.

P.A. Miranda and R.A. Garrido. Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. *Transportation Research Part E*, 40:183–207, 2004.

J.A. Muckstadt. *Analysis and Algorithms for Service Parts Supply Chains*. Springer, New York, NY, 2005.

J.A. Muckstadt. A model for multi-item, multi-echelon, multi-indenture inventory system. *Management Science*, 20:472–481, 1973.

J.A. Muckstadt and L.J. Thomas. Are multi-echelon inventory methods worth implementing in systems with low-demand-rate items? *Management Science*, 26:483–494, 1973.

L.K. Nozick. The fixed charge facility location problem with coverage restrictions. *Transportation Research Part E*, 37:281–296, 2001.

L.K. Nozick and M.A. Turnquist. Integrating inventory impacts into a fixed-charge model for locating distribution centers. *Transportation Research Part E*, 34(3):173–186, 1998.

L. Ozsen, M.S. Daskin, and C.R. Coullard. Capacitated facility location model with risk pooling. Technical report, Nortwestern University, 2004.

U. Pferschy. Dynamic programming revisited: Improving knapsack algorithms. *Computing*, 63(4):419–430, 1999.

K. Poole. Seizing the potential of the service supply chain. *Supply Chain Management Review*, pages 54–61, 2003.

J.A. Rappold and J.A. Muckstadt. A computationally efficient approach for determining inventory levels in a capacitated multiechelon production-distribution system. *Naval Research Logistics*, 47:377–398, 2000.

W.D. Rustenburg, G.J. van Houtum, and W.H.M. Zijm. Spare parts management at complex technology-based organizations: An agenda for research. *International Journal of Production Economics*, 71:177–193, 2001.

Z.-J.M. Shen and M.S. Daskin. Tradeoffs between customer service and cost in an integrated supply chain design framework. Submitted to Manufacturing and Service Operations Management, 2003.

Z.-J.M. Shen, C. Coullard, and M.S. Daskin. A joint location-inventory model. *Transportation Science*, 37(1):40–55, 2003.

C.C. Sherbrooke. METRIC: A multi-echelon technique for recoverable item control. *Operations Research*, 16:122–141, 1968.

C.C. Sherbrooke. VARI-METRIC: Improved approximation for multi-indenture, multi-echelon availability models. *Operations Research*, 34:311–319, 1986.

C.C. Sherbrooke. *Optimal Inventory Modeling of Systems*. Wiley, New York, NY, 1992.

L.V. Snyder. Facility location under uncertainty: A review. Technical report, Lehigh University, July 2004.

L.V. Snyder and M.S. Daskin. Reliability models for facility location: The expected failure cost case. Technical report, Lehigh University, April 2004.

L.V. Snyder, M.S. Daskin, and C.P. Teo. The stochastic location model with risk pooling. Technical report, Lehigh University, December 2003.

J.-S. Song. On the order fill rate in a multi-item, base-stock inventory system. *Operations Research*, 46(6), 1998.

C.-P. Teo, J. Ou, and M. Goh. Impact on inventory costs with consolidation of distribution centers. *IIE Transactions*, 32(2):99–110, 2001.

C.J. Vidal and M. Goetschalckx. Modeling the effect of uncertainties on global logistic systems. *Journal of Business Logistics*, 21(1), 2000.

W.E. Wilhelm. A technical review of column generation in integer programming. *Optimization and Engineering*, 2:159–200, 2001.

P. Zipkin. *Fundemantals of Inventory Management.* McGraw Hill - Irwin, Boston, MA, 2000.

Vita

Mehmet Ferhat Candas was born in Trabzon, Turkey on 5 November 1979, the son of Sevki and Sencan Candas. In 2002, he received his Bachelor of Science degree in Industrial Engineering from Bilkent University in Ankara, Turkey. Candas continued his studies as a graduate student in Operations Research and Industrial Engineering at the University of Texas at Austin in August 2002 where he pursued his doctoral studies.

Under the supervision of Dr. Erhan Kutanoglu, his Ph.D. dissertation research focuses on the integration of network design and inventory problems for service parts logistics systems using mathematical programming and combinatorial optimization. His research is supported by National Science Foundation and IBM.

He is currently working in the Global Manufacturing Systems department of AMD where he works to improve supporting tools for supply chain optimization and production planning. He is a member of IIE and INFORMS.

Permanent address: 3370 Lake Austin Blvd. Apt: D
Austin, Texas 78703

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.